Laboratoire d'Arithmétique, de Calcul formel et d'Optimisation
ESA - CNRS 6090

SKETCHES AND SPECIFICATIONS

USER'S GUIDE

Second part:
Mosaics for Implicit Specification

Dominique Duval & Christian Lair

Rapport de recherche n° 2000-04

Université de Limoges, 123 avenue Albert Thomas, 87060 Limoges Cedex
Tél. 05 55 45 73 23 - Fax. 05 55 45 73 22 - laco@unilim.fr

http://www.unilim.fr/laco/

SKETCHES AND SPECIFICATIONS

USER'S GUIDE

Second part:

Mosaics for Implicit Specification

Dominique DUVAL

Université de Limoges
Laboratoire d'Arithmétique, Calcul formel et Optimisation
123 avenue Albert Thomas, 87060 Limoges Cedex, France
dominique.duval@unilim.fr

and

Christian LAIR

Université Denis Diderot - Paris 7
U.F.R. de Mathématiques, Équipe Catégories et Structures
2 place Jussieu, 75251 Paris Cedex 05, France
lairchrist@aol.com

February 17, 2000

`http://www.unilim.fr/laco/rapports`

SKETCHES AND SPECIFICATIONS — USER'S GUIDE

SKETCHES AND SPECIFICATIONS is a common denomination for several papers which deal with applications of Ehresmann's sketch theory to computer science. These papers can be considered as the first steps towards a unified theory for software engineering. However, their aim is not to advocate a unification of computer languages; they are designed to build a frame for the study of notions which arise from several areas in computer science.

These papers are arranged in two complementary families:

REFERENCE MANUAL and USER'S GUIDE.

The *reference manual* provides general definitions and results, with comprehensive proofs. On the other hand, the *user's guide* places emphasis on motivations and gives a detailed description of several examples. These two families, though complementary, can be read independently. No prerequisite is assumed; however, it can prove helpful to be familiar either with specification techniques in computer science or with category theory in mathematics.

These papers are under development, they are, or will be, available at:
`http://www.unilim.fr/laco/rapports`.

REFERENCE MANUAL:

First Part: Compositive Graphs
Second Part: Projective Sketches
Third Part: Models

USER'S GUIDE:

First Part: Wefts for Explicit Specification
Second Part: Mosaics for Implicit Specification
Third Part: Functional and Imperative Programs

In addition, further papers about APPLICATIONS are in progress, with several co-authors. They deal with various topics, including the notion of state in computer science [state], overloading, coercions and subsorts.

These articles owe a great deal to the working group *sketches and computer algebra*; we would like to thank its participants, specially Catherine Oriat and Jean-Claude Reynaud, as well as the CNRS.

These papers have been processed with LaTeX and XY-pic.

Second Part: Mosaics for Implicit Specification

The aim of this paper is to introduce new specifications, more powerful than wefts. These new specifications are called *mosaics*. They generalize wefts, in order to be able to deal with implicit features of computer languages. In addition, each mosaic can be *explicited*, by building a weft with the same meaning. This construction is called the *ribbon product*, it is related to the family of tensor products.

This paper is part of a general study of some applications of Ehresmann's sketch theory to computer science, along with the reference manual [ref]. The only prerequisite to read this article is [guide1]. Here, as well as in [guide1], we focus on the interpretations of our specifications. The point of view of programs and computation will be studied in [guide3].

1 Introduction

In [guide1] we define wefts and some tools for handling them. Wefts have been introduced by Lair in [Lair 87]. They come from Ehresmann's theory of sketches [Ehresmann 66, Ehresmann 68] and they are related to *patchworks* [Lair 93]. They are also somewhat related to notions introduced by Freyd in [Freyd 73, Freyd & Scedrov 90]. Here is a short survey of [guide1].

- An $\mathcal{A}$-*weft*, for any category $\mathcal{A}$, is made of a *support* (which is a point of $\mathcal{A}$) and *constraints*. A *realization* ω of an $\mathcal{A}$-weft $\mathbf{S}$ towards a point A of $\mathcal{A}$ is *specified by* $\mathbf{S}$:

  $$\omega \in \mathrm{Real}_{\mathcal{A}}(\mathbf{S}, A) \ .$$

  When $\mathcal{A}$ is the category $\mathcal{A}mbi$ of *ambigraphs* and A is the ambigraph $\mathcal{S}et$ of *sets*, this notion of specification is very similar to the notion of *algebraic specification* [Goguen *et al.* 78, Astesiano *et al.* 99].

- A category $\mathcal{A}$ is *sketched* by a sketch $\mathbf{E}$ if $\mathcal{A}$ is equivalent to the category of *models* of $\mathbf{E}$:

  $$\mathcal{A} \simeq \mathcal{M}od(\mathbf{E}, \mathcal{S}et) \ .$$

  For example, the category $\mathcal{A}mbi$ is sketched by the projective sketch $\mathbf{E}_{\mathcal{A}mbi}$.

- Given a projective sketch $\mathbf{E}$, each model μ of $\mathbf{E}$ determines another projective sketch: the *blow-up* $\mathbf{E}\backslash\mu$ of $\mathbf{E}$ by μ. On the other hand, a μ-*indexation* is a homomorphism $h : \nu \to \mu$ of models of $\mathbf{E}$; it is a point of the category $\mathcal{M}od(\mathbf{E}, \mathcal{S}et)/\mu$ of μ-indexations. The *fundamental property of the blow-up* states that the category of models of $\mathbf{E}\backslash\mu$ is equivalent to the category of μ-indexations:

  $$\mathcal{M}od(\mathbf{E}, \mathcal{S}et)/\mu \simeq \mathcal{M}od(\mathbf{E}\backslash\mu, \mathcal{S}et) \ .$$

This paper is devoted to the definition of *mosaics* and their *realizations*, and to the study of a construction called the *ribbon product* which assigns to each mosaic a weft with the same realizations. A large part of this paper deals with an example. A comprehensive study of the ribbon product will be found in our reference manual. Several papers in progress deal with applications of these ideas to various implicit features of computer languages.

The only prerequisite to read this paper is [guide1].

Implicit features in computer languages include side effects, error handling, partiality, overloading, coercions,... They can be found in all computer languages, at various degrees. They are fundamental in *imperative* languages like Pascal or C, while they are of minor importance in *applicative* (or *functional*) languages like Lisp or ML. Unlike what happens with natural languages, the implicit features of computer languages should be made explicit before the program is run. If this is not done correctly, it may lead to surprising results.

Specification by wefts of ambigraphs, together with their set-valued realizations, is sufficient for dealing with purely functional programming. This will be seen in [guide3], and can be deduced from the important work done in (and around) the theory of algebraic specifications: a fundamental paper is [Goguen *et al.* 78], whereas [Astesiano *et al.* 99] is a recent survey. This works in the following way. A functional program can be defined as a *term* p of a weft of ambigraphs $\mathbf{S}$. A term of $\mathbf{S}$ is composed of arrows from the support of $\mathbf{S}$ and arrows which arise from the constraints of $\mathbf{S}$. The meaning of such a program is given by the interpretation of the

term p by a set-valued realization ω of $\mathbf{S}$. For instance, let $\mathbf{S}$ be the weft of ambigraphs made of two points N and U, two arrows $z : U \to N$ and $s : N \to N$, and the constraint $U = \mathbb{1}$. Let ω be the set-valued realization of $\mathbf{S}$ which interprets N, U, z and s respectively as $\mathbb{N}$, a one-element set $\mathbb{U} = \{*\}$, the constant map $* \mapsto 0$ (identified with the natural number 0) and the *successor* map $succ : n \mapsto n+1$. Then, the term $p = s \circ s \circ z : U \to N$ is a functional program, and its interpretation by ω is the constant map $* \mapsto 2$ (identified with the natural number 2).

Our aim is to generalize this result to languages with implicit features, like imperative languages. For this purpose, in this paper we define the *mosaics*, their *terms* and their *realizations*. As usual, the realizations give the meaning, while the terms determine the programs. However, the terms of a mosaic $\mathbf{\Sigma}$ are the terms of some weft $\mathbf{S}^a$, called *apparent*, whereas the realizations of $\mathbf{\Sigma}$ can be identified to the realizations of *another* weft $\mathbf{S}^e$, called *explicit*.

The construction of the explicit weft $\mathbf{S}^e$ from $\mathbf{\Sigma}$ enlightens the relation between the implicit and the explicit points of view on specifications (for example, specifications with implicit state or with explicit state, as in the example below). It also proves that the implicit point of view contains more information than the explicit one. The implicit point of view is also much better suited to the study of programs and computation: this will be seen in [guide3].

Let us now consider a typical example of imperative programming, including a notion of *state* and an *assignment* of a value to a *variable* (in the computer science meaning). Let I denote the instruction:

$$x := x + 1$$

where x is a variable of *type* (in the computer science meaning) natural integer.

It is well known [Dauchy & Gaudel 94] that the instruction I can be made into a functional program, at the price of some serious drawbacks. For this purpose, let us consider the following weft of ambigraphs $\mathbf{S}^e$ and its set-valued realization ω^e. The weft of ambigraphs $\mathbf{S}^e$ has two points N and E, which are interpreted respectively, by ω^e, as the set $\mathbb{N}$ of natural numbers and the set $\mathbb{E}$ of the states of the machine. Here the state is *explicit*, because it is represented by a point E in the weft. In addition, $\mathbf{S}^e$ has three arrows $s : N \to N$, $r : E \to E \times N$ and $u : E \times N \to E$, where s is interpreted, by ω^e, as the map *succ*, r as the map "read the value of x" and u as the map "update the value of x". Then the instruction I corresponds to the term $p^e = u \circ (id_E \times s) \circ r$ of $\mathbf{S}^e$:

$$E \xrightarrow{r} E \times N \xrightarrow{id_E \times s} E \times N \xrightarrow{u} E \,.$$

The interpretation of p^e is a map which assigns to each input state e_{in} in $\mathbb{E}$ an output state e_{out} in $\mathbb{E}$, where the value of x is incremented. However, this point of view becomes rapidly untractable because of the size of $\mathbf{S}^e$ and the occurrence of E nearly everywhere, see [Dauchy & Gaudel 94]. In addition, the programs cannot be identified with the terms: indeed the term p^e is fairly different from the instruction I. Another drawback is that, with this point of view, it is difficult to forbid adding to the weft the point $E \times E$, interpreted as $\mathbb{E}^2$, though this is generally not wished.

Actually, there is a second way to build a term corresponding to the instruction I. This point of view is most naive and does not seem to have any meaning. Looking at the instruction I in a formal way, without any attempt to understand its meaning, we get the following information: the symbol "x" to the right of I returns a value, which is a natural number, and does not need any argument; on the other hand, the symbol "$x :=$" to the left of I needs an argument of natural number type and does not return any value. Hence, let us consider the following weft of ambigraphs $\mathbf{S}^a$. It has a point N and an arrow $s : N \to N$, to be interpreted as the set $\mathbb{N}$

and the map *succ*, as well as a point U with the constraint $U = \mathbb{1}$ and two arrows $r^a : U \to N$ and $u^a : N \to U$, which are formal artefacts for translating respectively the symbols "x" and "$x :=$". Then the instruction I corresponds to the term $p^a = r^a \circ s \circ u^a$ of $\mathbf{S}^a$:

$$U \xrightarrow{r^a} N \xrightarrow{s} N \xrightarrow{u^a} U \ .$$

With this point of view the weft is smaller, the term is simpler, and in addition it preserves the shape of the instruction I. But it does not seem to have any meaning: indeed, in any set-valued realization ω of $\mathbf{S}^a$, the interpretation of U is a one-element set $\mathbb{U}$, and the interpretation of p^a is the unique map from $\mathbb{U}$ to $\mathbb{U}$, *i.e.* the identity of $\mathbb{U}$. This means that p^a does not change anything, whereas of course the instruction I does change something. We might look at realizations of $\mathbf{S}^a$ towards another ambigraph $\mathcal{V}$, but the result would be similar: the interpretation of U is a terminal point V_T of $\mathcal{V}$, hence the interpretation of p^a is the unique arrow with rank $V_T \to V_T$ in $\mathcal{V}$, namely the identity arrow of V_T.

However, we claim that:

- the term p^a can be interpreted (in a generalized sense) in order to recover the meaning of the instruction I;
- and it is possible to build the weft with explicit state $\mathbf{S}^e$ from $\mathbf{S}^a$, using a general construction called the *ribbon product*.

Moreover, we claim that a similar approach is valid for various implicit features of computer languages.

The generalized interpretation of p^a mentioned above, as well as the ribbon product construction, depends on the kind of implicit feature which is considered: in this example, it is an implicit notion of state. The information on this implicit feature can be given in various ways: for instance by syntactic conventions like the use of the assignment symbol ":=", or by informal comments which are added to the program: let us speak about *comments* in any case. In our example, comments deal with the state. They say that the interpretation of u^a may modify the current state, while the interpretation of r^a depends on the current state but is not allowed to modify it, and the interpretation of s should neither use nor modify it. We say that s is a *static* operation, r^a is an *access* operation, and u^a is a *modification* operation. In addition, we choose a name for each property, for instance here the names *st*, *acc* and *mod*, respectively. In this way, the information can be split between:

- the *apparent weft* $\mathbf{S}^a$,
- the description of the *properties*,
- the *names* of the properties: *st*, *acc* and *mod*,
- the *indexation* of the arrows of $\mathbf{S}^a$ by the property names:
  $s \mapsto st$, $r^a \mapsto acc$, $u^a \mapsto mod$.

This altogether is a (simplified) *mosaic* $\mathbf{\Sigma}$. The indexation in this example is a special case of a more general notion called a *stratification*.

$$\begin{tikzcd}
& \begin{array}{c}\text{property names}\\ \textit{ex.: } st,\ acc,\ mod\end{array} \arrow[dl] \arrow[dr] & & \begin{array}{c}\text{apparent weft}\\ \textit{ex.: } \mathbf{S}^a\end{array} \arrow[dl] \\
\begin{array}{c}\text{properties}\\ \textit{ex.: static, access, modification}\end{array} & & \begin{array}{c}\text{stratification}\\ \textit{ex.: } s \mapsto st,\ r^a \mapsto acc,\ u^a \mapsto mod\end{array} &
\end{tikzcd}$$

Our claim is that the mosaic $\mathbf{\Sigma}$ gives *at least as much* information as the weft $\mathbf{S}^e$. This will be proven by the construction of $\mathbf{S}^e$ from $\mathbf{\Sigma}$, using a ribbon product:

$$\mathbf{S}^e = \mathcal{E}xpl(\mathbf{\Sigma}) \ .$$

We call $\mathcal{E}xpl(\mathbf{\Sigma})$ the *explicit weft* associated with $\mathbf{\Sigma}$.

We will define the set $\mathrm{Real}(\mathbf{\Sigma}, \mathcal{V})$ of *realizations* of a mosaic $\mathbf{\Sigma}$ towards any ambigraph $\mathcal{V}$, directly from the components of $\mathbf{\Sigma}$ (properties and stratification). We will prove that these realizations are the same as the realizations of $\mathbf{S}^e$, by the *fundamental theorem on the ribbon product*:

$$\mathrm{Real}(\mathbf{\Sigma}, \mathcal{V}) = \mathrm{Real}(\mathcal{E}xpl(\mathbf{\Sigma}), \mathcal{V}) \ .$$

In the above example, this proves that the realizations of $\mathbf{\Sigma}$ yield the required meaning. This result states that implicit features can be made explicit, thanks to the weft $\mathcal{E}xpl(\mathbf{\Sigma})$, which is often quite large and intricate. This fact is well known from experience, here it becomes a theorem which is similar to the fundamental result on *tensor products* in category theory.

Actually, the mosaic $\mathbf{\Sigma}$ gives *more* information than the weft $\mathbf{S}^e$. For instance in our example, it can be proven that, with the mosaic, the use of the point $E \times E$ can be forbidden [guide3]. In fact, the components of $\mathbf{\Sigma}$ are all mixed together in $\mathbf{S}^e$, and can no longer be torn apart. These components split the information into three parts:

- on one hand, the apparent weft focuses on the computations involving one variable x of natural number type, without worrying about the state;
- on the other hand, the properties describe how an operation can interfere with the state;
- and both are related by the indexation, which says how each operation in the apparent weft does interfere with the state.

Modularity, *i.e.* building large specifications from small ones, is a fundamental process of any specification paradigm. Most usual specification constructors, like enrichments, extensions, quotients and pushouts [Goguen *et al.* 78, Wirsing 90], are special cases of the *inductive limit* (or *colimit*) constructor from category theory. The theorem which describes the realizations (often called models) of the new specification is the following one:

> *the realizations of the inductive limit of a family of specifications are the projective limits of the realizations of the given specifications.*

The ribbon product is a new tool for modularity, since it builds a large weft $\mathcal{E}xpl(\mathbf{\Sigma})$ from small pieces organized into a mosaic $\mathbf{\Sigma}$. The fundamental theorem describes the realizations of $\mathcal{E}xpl(\mathbf{\Sigma})$: they can be identified with the realizations of the mosaic $\mathbf{\Sigma}$, but *not* with the realizations of the apparent weft $\mathbf{S}$. Actually, in order to build a ribbon product, we use a *crown product*, which is a special kind of inductive limit. So, the fundamental theorem on the ribbon product is a consequence of the general theorem stated above about the realizations of an inductive limit of specifications.

Other methods have been brought forward for dealing with implicit features in computer languages, at least for dealing with the notion of state in an implicit way. Among them are *monads*, *linear logic*, and various methods involving a notion of *instant algebra*. We now give some hints about a comparison with our point of view.

The method using *monads* [Moggi 91, Wadler 85] makes use of an algebraic specification for dealing with the functional features of the language and monads for dealing with its implicit features. A generalization of this method can be found in [Filliatre 99]. Monads are related to a special kind of mosaic, as we will see in an example in 2.

The *linear logic* approach [Lafont 88, Wadler 90] is able to deal with the notion of state in such a way that the state cannot be duplicated (this problem has been mentioned above). This can also be obtained with a mosaic, thanks to a suitable definition of imperative programs [guide3].

The notion of *instant algebra* is crucial for the "state-as-algebra" approaches, like *dynamic systems* [Gaudel *et al.* 99, Lellahi & Zamulin 99], *D-oids* [Astesiano & Zucca 95], *dynamic abstract data types* [Ehrig & Orejas 94, Ehrig & Orejas 98] and *abstract state machines* (or *evolving algebras*) [Gurevich 99, Gurevich 91]. Let us use the terminology of [Ehrig & Orejas 94]. There are four specification levels, and the first three levels correspond to an increasingly large part of our apparent weft. The first level specifies the *values types*, and the corresponding part of our apparent weft contains all the arrows indexed by st. The second level specifies the *instant structures*, and it corresponds to adding the arrows indexed by acc. The third level, with *dynamic operations*, corresponds to the whole apparent weft, by adding the arrows indexed by mod.

Other points of view, such as those which rely upon *automata* theory [Hopcroft & Ullman 79], are designed for dealing with states, but it is difficult for them to deal with complex data types. With our point of view we may consider the usual graphic representations of automata as wefts, so that we may conjecture that our approach encompasses part of automata theory, along with algebraic specifications theory.

The *crown product* is introduced in 3, while the *stratifications*, the *mosaics* and the *ribbon product* are studied in 4. Sections 2 and 5 are devoted to a detailed study of two versions of an example. The aim of this example is to specify the natural numbers with a *predecessor* operation, in such a way that any attempt to evaluate the predecessor of zero results in an *error*. In both versions, the apparent weft takes care of the natural numbers while the properties deal with the treatment of errors. It would be easy to analyse other examples involving error handling, using the same properties but changing the apparent weft and the stratification. Our first version (in 2), using the crown product in a naive way, is not powerful enough. Our second version (in 5) gives a much better result, thanks to a mosaic and a ribbon product.

2 Crown product: an example

Examples here and in section 5 address the issue of computing with *natural numbers with a predecessor operation*, as described now.

The set $\mathbb{N}$ of natural numbers is freely generated by the element 0 and the successor map $succ : \mathbb{N} \to \mathbb{N}$. We wish to add a predecessor map such that $pred(succ(n)) = n$ for all $n \in \mathbb{N}$, and such that any attempt to compute the predecessor of 0 returns an error. In addition, in order to keep them simple, the programs should be written without any mention of a potential error: error handling should remain implicit.

This section is devoted to the study of a first way to deal with this example, which, as we will see, is not wholly satisfactory. However, this method leads to the introduction of the *crown product* in 3; then the crown product is used in 4 to define the *ribbon product*; finally, thanks to the ribbon product, a much better way to deal with this example is given in 5.

In 2.1, an analysis of the example leads to a weft of ambigraphs **S** which does not mention the error: it can be used to build programs, but its set-valued realizations are irrelevant. In order to get more relevant realizations we can try:

1. either to look at non-set-valued realizations of **S**,
2. or to look at set-valued realizations of another weft of ambigraphs, which might be built from **S** using a *crown product*.

Both points of view are studied here, first in 2.7 for the support of **S**, then in 2.8 and 2.9 for its constraints. Since the set-valued realizations of **S** are irrelevant, the first thing to do is to say precisely how each ingredient of **S** should be interpreted. This is done in 2.1 and 2.2.

2.1 Analysis

Let us now give a precise statement for our example.

The predecessor of 0 is an error, *i.e.* an element $\varepsilon_{\mathbb{N}}$ which is not a natural number. Hence the predecessor is $pred : \mathbb{N} \to \mathbb{N}'$, where $\mathbb{N}' = \mathbb{N} \sqcup \{\varepsilon_{\mathbb{N}}\}$, with the symbol "$\sqcup$" denoting the disjoint union, and $pred(0) = \varepsilon_{\mathbb{N}}$. Moreover, in order to propagate the error when maps are composed, the successor map $succ : \mathbb{N} \to \mathbb{N}$ is extended to a map $succ' : \mathbb{N}' \to \mathbb{N}'$ such that $succ'(\varepsilon_{\mathbb{N}}) = \varepsilon_{\mathbb{N}}$. In the same way, $pred : \mathbb{N} \to \mathbb{N}'$ is extended to $pred' : \mathbb{N}' \to \mathbb{N}'$ such that $pred'(\varepsilon_{\mathbb{N}}) = \varepsilon_{\mathbb{N}}$.

Now, as in [guide1], let $\mathbb{U} = \{*\}$ denote a one-element set, and 0 the constant map $* \mapsto 0 : \mathbb{U} \to \mathbb{N}$. The unique map $fact_{\mathbb{N}} : \mathbb{N} \to \mathbb{U}$, such that $n \mapsto *$ for all $n \in \mathbb{N}$, can also be useful. Hence, it should propagate the error, which means that it should be extended to $fact'_{\mathbb{N}} : \mathbb{N}' \to \mathbb{U}'$ where $\mathbb{U}' = \mathbb{U} \sqcup \{\varepsilon_{\mathbb{U}}\} = \{*, \varepsilon_{\mathbb{U}}\}$, in such a way that $fact'_{\mathbb{N}}(\varepsilon_{\mathbb{N}}) = \varepsilon_{\mathbb{U}}$. Then the constant map $0 : \mathbb{U} \to \mathbb{N}$ should also be extended, to $0' : \mathbb{U}' \to \mathbb{N}'$ such that $0'(\varepsilon_{\mathbb{U}}) = \varepsilon_{\mathbb{N}}$.

$$\begin{tikzcd}
\mathbb{U} \arrow[r, hook] \arrow[d, "0"] & \mathbb{U}' = \mathbb{U} \sqcup \{\varepsilon_{\mathbb{U}}\} \arrow[d, "0'"] & \{\varepsilon_{\mathbb{U}}\} \arrow[l, hook'] \\
\mathbb{N} \arrow[r, hook] \arrow[r, shift right, "pred"'] \arrow[loop below, "succ"] & \mathbb{N}' = \mathbb{N} \sqcup \{\varepsilon_{\mathbb{N}}\} \arrow[loop below left, "succ'"] \arrow[loop below right, "pred'"] & \{\varepsilon_{\mathbb{N}}\} \arrow[l, hook']
\end{tikzcd}$$

To sum up, each set X gives rise to a set $X' = X \sqcup \{\varepsilon_X\}$, disjoint union of X and of an element ε_X called *the error*, and each map $f : X \to Y$ (*resp.* $f : X \to Y'$) gives rise to a map $f' : X' \to Y'$ which extends f and which *propagates the error*, which means that $f'(\varepsilon_X) = \varepsilon_Y$.

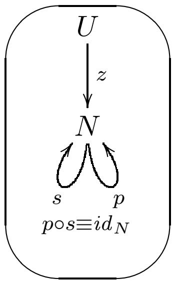
Now, let us look at the programs: they should be written without mentioning any error. This means that they should be written as if the *predecessor* map had all its values in $\mathbb{N}$. So that a program can be defined as a term of the following weft of ambigraphs **S**:

$\mathcal{A}mbi$-Weft **S**:

points:	U , N ,
arrows:	$z : U \to N$, $s : N \to N$, $p : N \to N$,
equation:	$p \circ s \equiv id_N$,
terminal point:	U ,
initiality constraint:	the interpretation of $(U \xrightarrow{z} N \xrightarrow{s} N)$ is initial among the $(X_0 \xrightarrow{x_0} X \xrightarrow{f} X)$ where X_0 is terminal.

the support of **S** is:

$\underline{\mathbf{S}}$:



and its constraints are described in [guide1, section 2.4].

The description of **S** is given here in its abbreviated form. Because of the equation, **S** also includes the identity arrow $id_N : N \to N$ and the composed arrow $p \circ s : N \to N$.

In a program, the arrows s and p can be composed at will: indeed $s \circ p \circ z$ is a term of **S** as well as $p \circ s \circ z$. Independently, the error handling should be such that $s \circ p \circ z$ is recognized as erroneous, whereas $p \circ s \circ z$ is not.

Let us check that the set-valued realizations of **S** are irrelevant. Each set-valued realization of **S** interprets U, N, z and $\widetilde{s}$ respectively as $\mathbb{U}$, $\mathbb{N}$, $\widetilde{0}$ and *succ*, because of the constraints. It also interprets p as a map $\widetilde{pred} : \mathbb{N} \to \mathbb{N}$ such that $\widetilde{pred}(n+1) = n$, for all $n \in \mathbb{N}$, because of the equation. Hence the value of $\widetilde{pred}(0)$ is a natural number, instead of an error.

Since the set:

$$\mathrm{Real}_{\mathcal{A}mbi}(\mathbf{S}, \mathcal{S}et)$$

of set-valued realizations of **S** is irrelevant, we are going to build two other sets of realizations which are much more relevant:

1. on one hand, the set of realizations of **S** in an ambigraph different from $\mathcal{S}et$:

$$\mathrm{Real}_{\mathcal{A}mbi}(\mathbf{S}, \ldots) ,$$

2. on the other hand, the set of set-valued realizations of a weft of ambigraphs different from **S**, obtained from **S** by a *crown product*:

$$\mathrm{Real}_{\mathcal{A}mbi}(\ldots \copyright\, \mathbf{S}, \mathcal{S}et) .$$

As we will see, both points of view lead to the same result, but this result is not wholly satisfactory. It is only in 5, thanks to a more subtle approach, that we will get a good answer to the question of dealing with natural numbers with a predecessor operation.

2.2 Comments

It follows from 2.1 that we do not look at the set-valued realizations of **S**. Instead, we look at some *interpretations* (in a sense which has to be defined) of **S**, which satisfy the following *comments.*

The interpretation of each point of **S** should satisfy the comment:

- $K(\mathtt{Pt})$: the interpretation of a point G satisfies $K(\mathtt{Pt})$ if it is made of a set X' which *contains an error element* (denoted ε_X). Let X denote the complement of $\{\varepsilon_X\}$ in X', so that $X'=X \sqcup \{\varepsilon_X\}$.

The interpretation of each arrow of **S** should satisfy the comment:

- $K(\mathtt{Ar})$: the interpretation of an arrow $g : G_1 \to G_2$ satisfies $K(\mathtt{Ar})$ (knowing that the interpretation of the points G_1 and G_2 satisfies $K(\mathtt{Pt})$) if it is made of a map $f' : X'_1 \to X'_2$ which *propagates the error*, which means that $f'(\varepsilon_1)=\varepsilon_2$.

It can happen for these comments themselves to be expressed by means of wefts. This will be seen in 2.6, but let us first introduce some notations.

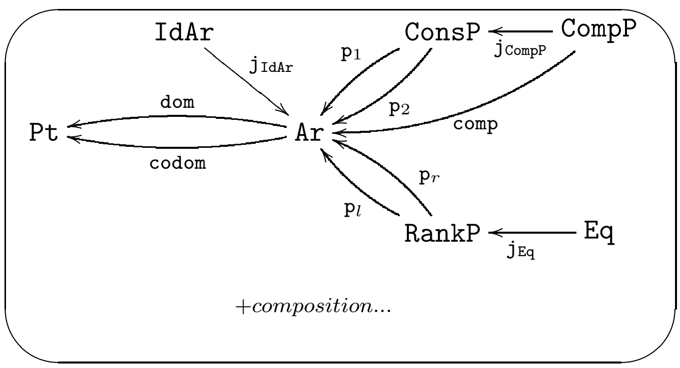
2.3 Projective sketch E

Let:

$$\mathbf{E}=\mathbf{E}_{\mathcal{A}mbi}$$

denote the projective sketch of ambigraphs as in [guide1, section 3.4], with support:

$\underline{\mathbf{E}}$:

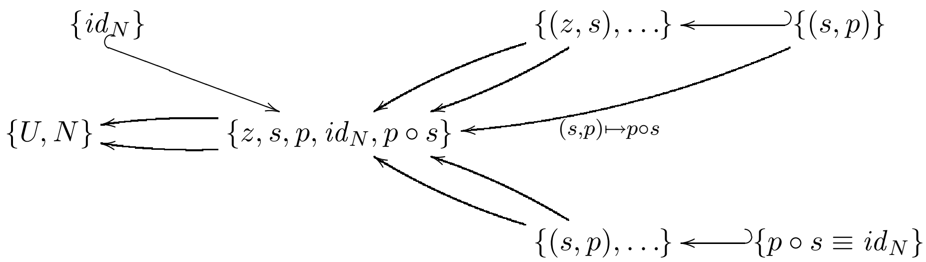


2.4 Model $\mu : \mathbf{E} \to \mathcal{S}et$

First let us look at the support of **S**. Its constraints will be studied in 2.8 and in 2.9 respectively. Let:

$$\mu=\underline{\mathbf{S}} : \mathbf{E} \to \mathcal{S}et\ .$$

This is an ambigraph, identified with a model of the projective sketch **E**:



2.5 Contravariant functor $\Theta : \mathcal{W} \longrightarrow \mathcal{S}et$

Let $\mathcal{W}$ denote the category of wefts of ambigraphs [guide1, sections 2.1and 2.4]:

$$\mathcal{W} = \mathcal{W}\!\mathit{eft}(\mathcal{A}) \quad \text{where} \quad \mathcal{A} = \mathcal{A}\!\mathit{mbi}\,,$$

and let Θ denote the contravariant functor:

$$\Theta = \mathrm{Real}_{\mathcal{A}}(-, \mathcal{S}et) : \mathcal{W} \longrightarrow \mathcal{S}et\,.$$

It is left-exact, which means that it maps each inductive limit cone of $\mathcal{W}$ to a projective limit cone of $\mathcal{S}et$.

2.6 Counter-model $\kappa : \mathbf{E} \longrightarrow \mathcal{W}$

From 2.2, the interpretation of each point of μ should satisfy the comment $K(\mathtt{Pt})$, and the interpretation of each arrow of μ should satisfy the comment $K(\mathtt{Ar})$. In both cases, *i.e.* for $E = \mathtt{Pt}$ as for $E = \mathtt{Ar}$, the comment $K(E)$ *corresponds* to a weft of ambigraphs $\kappa(E)$: the interpretations which satisfy the comment $K(E)$ can be identified to the set-valued realizations of $\kappa(E)$.

$\mathcal{A}$-WEFT $\kappa(\mathtt{Pt})$:
- *points*: $H\,,\ H'\,,\ H^e\,,$
- *arrows*: $h : H \to H'\,,\ h^e : H^e \to H'\,,$
- *terminal point*: $H^e\,,$
- *sum*: $H \xrightarrow{h} H' \xleftarrow{h^e} H^e\,.$

$$\kappa(\mathtt{Pt}) : \quad H \xrightarrow{\ h\ } H' = H + H^e \xleftarrow{\ h^e\ } H^e = \mathbb{1}$$

Let us check that the comment $K(\mathtt{Pt})$ corresponds to the weft of ambigraphs $\kappa(\mathtt{Pt})$: let ω be a set-valued realization of $\kappa(\mathtt{Pt})$, and let $X = \omega(H)$, $X' = \omega(H')$, and ε_X be the unique element of $\omega(h^e)(\omega(H^e))$. We then get $X' = X \sqcup \{\varepsilon_X\}$.

In the description of $\kappa(\mathtt{Ar})$, and everywhere below, referring to our drawings, the "*line* $\kappa(\mathtt{Pt})_n$" stands for a copy of $\kappa(\mathtt{Pt})$ with the names of all its ingredients indexed by the symbol n. This means that the line $\kappa(\mathtt{Pt})_n$ is the weft of ambigraphs with support $(H_n \xrightarrow{h_n} H'_n \xleftarrow{h^e_n} H^e_n)$ and constraints $H^e_n = \mathbb{1}$ and $H'_n = H_n + H^e_n$.

$\mathcal{A}$-WEFT $\kappa(\mathtt{Ar})$:
- *lines*: $\kappa(\mathtt{Pt})_1\,,\ \kappa(\mathtt{Pt})_2\,,$
- *arrows*: $k' : H'_1 \to H'_2\,,\ k^e : H^e_1 \to H^e_2\,,$
- *equation*: $k' \circ h^e_1 \equiv h^e_2 \circ k^e\,.$

$$\kappa(\mathtt{Ar}) : \quad \begin{tikzcd}
H_1 \arrow[r, "h_1"] & H'_1 = H_1 + H^e_1 \arrow[d, "k'"] \arrow[r, phantom, "\equiv" below=12pt] & H^e_1 = \mathbb{1} \arrow[l, "h^e_1"'] \arrow[d, "k^e"] \\
H_2 \arrow[r, "h_2"] & H'_2 = H_2 + H^e_2 & H^e_2 = \mathbb{1} \arrow[l, "h^e_2"']
\end{tikzcd}$$

The comment $K(\mathtt{Ar})$ corresponds to the weft of ambigraphs $\kappa(\mathtt{Ar})$: indeed, let ω be a set-valued realization of $\kappa(\mathtt{Ar})$, then the map $\omega(k') : \omega(H'_1) \to \omega(H'_2)$ propagates the error.

Now it is easy to get a contravariant functor κ from $\underline{\mathbf{E}}$ towards the category $\mathcal{W}$:

$$\begin{tikzcd}
 & \kappa(\mathtt{IdAr}) & & \kappa(\mathtt{ConsP}) \arrow[r, "\kappa(\mathtt{j}_{\mathtt{CompP}})"'] & \kappa(\mathtt{CompP}) \\
\kappa(\mathtt{Pt}) \arrow[rr, bend left, "\kappa(\mathtt{dom})"] \arrow[rr, bend right, "\kappa(\mathtt{codom})"'] & & \kappa(\mathtt{Ar}) \arrow[ul, "\kappa(\mathtt{j}_{\mathtt{IdAr}})"'] \arrow[ur, bend left, "\kappa(\mathtt{p}_1)"] \arrow[ur, bend right, "\kappa(\mathtt{p}_2)"'] \arrow[urr, "\kappa(\mathtt{comp})"'] \arrow[dr, bend left, "\kappa(\mathtt{p}_r)"] \arrow[dr, bend right, "\kappa(\mathtt{p}_l)"'] & & \\
 & & & \kappa(\mathtt{RankP}) \arrow[r, "\kappa(\mathtt{j}_{\mathtt{Eq}})"'] & \kappa(\mathtt{Eq})
\end{tikzcd}$$

Homomorphism $\kappa(\mathtt{dom}) : \kappa(\mathtt{Pt}) \to \kappa(\mathtt{Ar})$:
lines: $\kappa(\mathtt{Pt}) \mapsto \kappa(\mathtt{Pt})_1$.

Homomorphism $\kappa(\mathtt{codom}) : \kappa(\mathtt{Pt}) \to \kappa(\mathtt{Ar})$:
lines: $\kappa(\mathtt{Pt}) \mapsto \kappa(\mathtt{Pt})_2$.

$\mathcal{A}$-Weft $\kappa(\mathtt{IdAr})$:
extends: $\kappa(\mathtt{Pt})$,
identity arrows: $id_{H'} : H' \to H'$, $id_{H^e} : H^e \to H^e$,
equation: $id_{H'} \circ h^e \equiv h^e \circ id_{H^e}$.

$$\kappa(\mathtt{IdAr}) : \quad \begin{tikzcd}
H \arrow[r, "h"] & H' = H + H^e \arrow[loop above, "id_{H'}"] \arrow[r, phantom, "\equiv"] & H^e = \mathbb{1} \arrow[l, "h^e"'] \arrow[loop above, "id_{H^e}"]
\end{tikzcd}$$

Homomorphism $\kappa(\mathtt{j}_{\mathtt{IdAr}}) : \kappa(\mathtt{Ar}) \to \kappa(\mathtt{IdAr})$:
lines: $\kappa(\mathtt{Pt})_1 \mapsto \kappa(\mathtt{Pt})$, $\kappa(\mathtt{Pt})_2 \mapsto \kappa(\mathtt{Pt})$,
arrows: $k' \mapsto id_{H'}$, $k^e \mapsto id_{H^e}$.

$\mathcal{A}$-Weft $\kappa(\mathtt{ConsP})$:
lines: $\kappa(\mathtt{Pt})_1$, $\kappa(\mathtt{Pt})_2$, $\kappa(\mathtt{Pt})_3$,
arrows: $k'_1 : H'_1 \to H'_2$, $k^e_1 : H^e_1 \to H^e_2$, $k'_2 : H'_2 \to H'_3$, $k^e_2 : H^e_2 \to H^e_3$,
equations: $k'_1 \circ h^e_1 \equiv h^e_2 \circ k^e_1$, $k'_2 \circ h^e_2 \equiv h^e_3 \circ k^e_2$.

$$\kappa(\mathtt{ConsP}) : \quad \begin{tikzcd}
H_1 \arrow[r, "h_1"] & H'_1 = H_1 + H^e_1 \arrow[d, "k'_1"] \arrow[dr, phantom, "\equiv"] & H^e_1 = \mathbb{1} \arrow[l, "h^e_1"'] \arrow[d, "k^e_1"] \\
H_2 \arrow[r, "h_2"] & H'_2 = H_2 + H^e_2 \arrow[d, "k'_2"] \arrow[dr, phantom, "\equiv"] & H^e_2 = \mathbb{1} \arrow[l, "h^e_2"'] \arrow[d, "k^e_2"] \\
H_3 \arrow[r, "h_3"] & H'_3 = H_3 + H^e_3 & H^e_3 = \mathbb{1} \arrow[l, "h^e_3"']
\end{tikzcd}$$

Homomorphism $\kappa(\mathtt{p}_1) : \kappa(\mathtt{Ar}) \to \kappa(\mathtt{ConsP})$:
lines: $\kappa(\mathtt{Pt})_1 \mapsto \kappa(\mathtt{Pt})_1$, $\kappa(\mathtt{Pt})_2 \mapsto \kappa(\mathtt{Pt})_2$,
arrows: $k' \mapsto k'_1$, $k^e \mapsto k^e_1$.

HOMOMORPHISM $\kappa(\mathtt{p}_2) : \kappa(\mathtt{Ar}) \to \kappa(\mathtt{ConsP})$:
lines: $\kappa(\mathtt{Pt})_1 \mapsto \kappa(\mathtt{Pt})_2$, $\kappa(\mathtt{Pt})_2 \mapsto \kappa(\mathtt{Pt})_3$,
arrows: $k' \mapsto k_2'$, $k^e \mapsto k_2^e$.

$\mathcal{A}$-WEFT $\kappa(\mathtt{CompP})$:
extends: $\kappa(\mathtt{ConsP})$,
arrows: $k' : H_1' \to H_3'$, $k^e : H_1^e \to H_3^e$,
composition: $k' = k_2' \circ k_1'$, $k^e = k_2^e \circ k_1^e$,
equation: $k' \circ h_1^e \equiv h_3^e \circ k^e$.

$$\kappa(\mathtt{CompP}) : \begin{tikzcd} H_1 \arrow[r, "h_1"] & H_1' = H_1 + H_1^e \arrow[d, "k_1'"] \arrow[dd, bend right, "k'"'] \arrow[dr, phantom, "\equiv"] & H_1^e = \mathbb{1} \arrow[l, "h_1^e"'] \arrow[d, "k_1^e"] \arrow[dd, bend right, "k^e"'] \\ H_2 \arrow[r, "h_2"] & H_2' = H_2 + H_2^e \arrow[d, "k_2'"] \arrow[dr, phantom, "\equiv"] & H_2^e = \mathbb{1} \arrow[l, "h_2^e"'] \arrow[d, "k_2^e"] \\ H_3 \arrow[r, "h_3"] & H_3' = H_3 + H_3^e & H_3^e = \mathbb{1} \arrow[l, "h_3^e"'] \end{tikzcd}$$

HOMOMORPHISM $\kappa(\mathtt{j}_{\mathtt{CompP}}) : \kappa(\mathtt{ConsP}) \to \kappa(\mathtt{CompP})$:
this is the extension.

HOMOMORPHISM $\kappa(\mathtt{comp}) : \kappa(\mathtt{Ar}) \to \kappa(\mathtt{CompP})$:
lines: $\kappa(\mathtt{Pt})_1 \mapsto \kappa(\mathtt{Pt})_1$, $\kappa(\mathtt{Pt})_2 \mapsto \kappa(\mathtt{Pt})_3$,
arrows: $k' \mapsto k'$, $k^e \mapsto k^e$.

$\mathcal{A}$-WEFT $\kappa(\mathtt{RankP})$:
lines: $\kappa(\mathtt{Pt})_1$, $\kappa(\mathtt{Pt})_2$,
arrows: $k_g' : H_1' \to H_2'$, $k_g^e : H_1^e \to H_2^e$, $k_d' : H_1' \to H_2'$, $k_d^e : H_1^e \to H_2^e$,
equations: $k_g' \circ h_1^e \equiv h_2^e \circ k_g^e$, $k_d' \circ h_1^e \equiv h_2^e \circ k_d^e$.

$$\kappa(\mathtt{RankP}) : \begin{tikzcd} H_1 \arrow[r, "h_1"] & H_1' = H_1 + H_1^e \arrow[d, bend right, "k_g'"'] \arrow[d, bend left, "k_d'"] \arrow[dr, phantom, "\equiv\equiv"] & H_1^e = \mathbb{1} \arrow[l, "h_1^e"'] \arrow[d, bend right, "k_g^e"'] \arrow[d, bend left, "k_d^e"] \\ H_2 \arrow[r, "h_2"] & H_2' = H_2 + H_2^e & H_2^e = \mathbb{1} \arrow[l, "h_2^e"'] \end{tikzcd}$$

HOMOMORPHISM $\kappa(\mathtt{p}_g) : \kappa(\mathtt{Ar}) \to \kappa(\mathtt{RankP})$:
lines: $\kappa(\mathtt{Pt})_1 \mapsto \kappa(\mathtt{Pt})_1$, $\kappa(\mathtt{Pt})_2 \mapsto \kappa(\mathtt{Pt})_2$,
arrows: $k' \mapsto k_g'$, $k^e \mapsto k_g^e$.

HOMOMORPHISM $\kappa(\mathtt{p}_d) : \kappa(\mathtt{Ar}) \to \kappa(\mathtt{RankP})$:
lines: $\kappa(\mathtt{Pt})_1 \mapsto \kappa(\mathtt{Pt})_1$, $\kappa(\mathtt{Pt})_2 \mapsto \kappa(\mathtt{Pt})_2$,
arrows: $k' \mapsto k_d'$, $k^e \mapsto k_d^e$.

$\mathcal{A}$-WEFT $\kappa(\mathtt{Eq})$:

extends: $\kappa(\texttt{RankP})$,
equations: $k'_g \equiv k'_d,\ k^e_g \equiv k^e_d$.

$$\kappa(\texttt{Eq}) : \quad \begin{tikzcd} H_1 \arrow[r, "h_1"] & H'_1 = H_1 + H^e_1 \arrow[d, shift right, "k'_g"'] \arrow[d, shift left, "k'_d"] \arrow[d, phantom, "\equiv"] \arrow[r, phantom, "\equiv\equiv"] & H^e_1 = \mathbb{1} \arrow[l, "h^e_1"'] \arrow[d, shift right, "k^e_g"'] \arrow[d, shift left, "k^e_d"] \arrow[d, phantom, "\equiv"] \\ H_2 \arrow[r, "h_2"] & H'_2 = H_2 + H^e_2 & H^e_2 = \mathbb{1} \arrow[l, "h^e_2"'] \end{tikzcd}$$

HOMOMORPHISM $\kappa(\texttt{j}_{\texttt{Eq}}) : \kappa(\texttt{RankP}) \to \kappa(\texttt{Eq})$:
this is the extension.

In this way we get a contravariant functor $\kappa : \underline{\mathbf{E}} \;+\!\!\longrightarrow\; \mathcal{W}$. Moreover, this functor maps each of the five distinguished projective cones of $\mathbf{E}$ to an inductive limit cone, so that it is a counter-model:

$$\kappa : \mathbf{E} \;+\!\!\longrightarrow\; \mathcal{W} \ .$$

2.7 Crown product $\kappa \,©_{\mathbf{E}}\, \mu$

We have just said that the interpretations of the points U and N of μ should be set-valued realizations of $\kappa(\texttt{Pt})$, *i.e.* elements of the set $\mathrm{Real}_{\mathcal{A}}(\kappa(\texttt{Pt}), \mathcal{S}et)$. Similarly, the interpretations of the arrows z, s, p of μ should be set-valued realizations of $\kappa(\texttt{Ar})$, *i.e.* elements of the set $\mathrm{Real}_{\mathcal{A}}(\kappa(\texttt{Ar}), \mathcal{S}et)$. In addition, it is easy to check that the interpretation of the identity arrow id_N should be a set-valued realization of $\kappa(\texttt{IdAr})$, that the interpretation of the pair of composable arrows (s, p) should be a set-valued realization of $\kappa(\texttt{CompP})$, that the interpretation of the equation $p \circ s \equiv id_N$ should be a set-valued realization of $\kappa(\texttt{Eq})$, and so on. To sum up:

- the interpretation of an ingredient of μ of nature E (for all point E of $\mathbf{E}$) should be an element of the set $\mathrm{Real}_{\mathcal{A}}(\kappa(E), \mathcal{S}et)$.

Now this study can be carried on in two ways:

1. The composition of the counter-model $\kappa : \mathbf{E} \;+\!\!\longrightarrow\; \mathcal{W}$ and the contravariant functor Θ : $\mathcal{W} \;+\!\!\longrightarrow\; \mathcal{S}et$ (which is left-exact) is a model of $\mathbf{E}$:

$$\Theta \circ \kappa = \mathrm{Real}_{\mathcal{A}}(\kappa(-), \mathcal{S}et) : \mathbf{E} \to \mathcal{S}et \ .$$

It interprets each point E of $\mathbf{E}$ as the set $\mathrm{Real}_{\mathcal{A}}(\kappa(E), \mathcal{S}et)$ of set-valued realizations of the weft of ambigraphs $\kappa(E)$.

$$\begin{tikzcd} & \mathbf{E} \arrow[dl, "\kappa"'] \arrow[d, shift right, "\Theta\circ\kappa"'] \arrow[d, shift left, "\mu"] \\ \mathcal{W} \arrow[r, "\Theta"'] \arrow[ur, phantom, "="] & \mathcal{S}et \end{tikzcd}$$

Since $\Theta \circ \kappa$ is a model of $\mathbf{E}$, it is an ambigraph. From the previous study, the natural numbers with the predecessor operation define a homomorphism from μ to $\Theta \circ \kappa$, *i.e.* an element of the set:

$$\boxed{\mathrm{Hom}_{\mathcal{M}od(\mathbf{E})}(\mu, \Theta \circ \kappa) \ .}$$

Precisely:
points: $U \mapsto \mathbb{U}' = \mathbb{U} + \{\varepsilon_{\mathbb{U}}\}$, $N \mapsto \mathbb{N}' = \mathbb{N} + \{\varepsilon_{\mathbb{N}}\}$,
arrows: $z \mapsto 0'$, $s \mapsto succ'$, $p \mapsto pred'$.

2. Here is another point of view. Since we would like to interpret the point U of μ as a set-valued realization of $\kappa(\mathtt{Pt})$, we replace in μ the point U by a copy of $\kappa(\mathtt{Pt})$. We do the same for the point N. Then we replace each arrow z, s and p of μ by a copy of $\kappa(\mathtt{Ar})$, and so on, for all the ingredients of μ. In this way we get a weft of ambigraphs which is called the *crown product* of κ by μ above $\mathbf{E}$, and is denoted:

$$\kappa \copyright_{\mathbf{E}} \mu \; .$$

$\mathcal{A}$-WEFT $\kappa \copyright_{\mathbf{E}} \mu$:
lines: $\kappa(\mathtt{Pt})_U$, $\kappa(\mathtt{Pt})_N$,
arrows: $k'_z : H'_U \to H'_N$, $k^e_z : H^e_U \to H^e_N$,
$k'_s : H'_N \to H'_N$, $k^e_s : H^e_N \to H^e_N$,
$k'_p : H'_N \to H'_N$, $k^e_p : H^e_N \to H^e_N$,
equations: $k'_z \circ h^e_U \equiv h^e_N \circ k^e_z$,
$k'_s \circ h^e_N \equiv h^e_N \circ k^e_s$,
$k'_p \circ h^e_N \equiv h^e_N \circ k^e_p$,
$k'_p \circ k'_s \equiv id_{H'_N}$, $k^e_p \circ k^e_s \equiv id_{H^e_N}$.

$$\kappa \copyright_{\mathbf{E}} \mu : \quad \begin{tikzcd} H_U \arrow[r, "h_U"] & H'_U = H_U + H^e_U \arrow[d, "k'_z"'] & H^e_U = \mathbb{1} \arrow[l, "h^e_U"'] \arrow[d, "k^e_z"] \\ H_N \arrow[r, "h_N"] & H'_N = H_N + H^e_N \arrow[loop below, "k'_s"'] \arrow[loop below, "k'_p"] & H^e_N = \mathbb{1} \arrow[l, "h^e_N"'] \arrow[loop below, "k^e_s"'] \arrow[loop below, "k^e_p"] \end{tikzcd}$$

+equations...

From the previous study, the natural numbers with the predecessor operation define a set-valued realization of $\kappa \copyright_{\mathbf{E}} \mu$, *i.e.* an element of the set:

$$\boxed{\Theta(\kappa \copyright_{\mathbf{E}} \mu) \; .}$$

Precisely:
points: $H_U \mapsto \mathbb{U}$, $H'_U \mapsto \mathbb{U}'$, $H^e_U \mapsto \{\varepsilon_{\mathbb{U}}\}$,
$H_N \mapsto \mathbb{N}$, $H'_N \mapsto \mathbb{N}'$, $H^e_N \mapsto \{\varepsilon_{\mathbb{N}}\}$,
arrows: $k'_z \mapsto 0'$, $k'_s \mapsto succ'$, $k'_p \mapsto pred'$,
$k^e_z \mapsto (\varepsilon_{\mathbb{U}} \mapsto \varepsilon_{\mathbb{N}})$, $k^e_s \mapsto id_{\{\varepsilon_{\mathbb{N}}\}}$, $k^e_p \mapsto id_{\{\varepsilon_{\mathbb{N}}\}}$.

Both these points of view are equivalent: indeed, proposition 1 will prove that there is a canonical bijection:

$$\boxed{\Theta(\kappa \copyright_{\mathbf{E}} \mu) \cong \mathrm{Hom}_{\mathcal{M}od(\mathbf{E})}(\mu, \Theta \circ \kappa) \; ,}$$

which means that:

$$\mathrm{Real}_{\mathcal{A}mbi}(\kappa \copyright_{\mathbf{E}_{\mathcal{A}mbi}} \mu, \mathcal{S}et) \cong \mathrm{Hom}_{\mathcal{A}mbi}(\mu, \mathrm{Real}_{\mathcal{A}mbi}(\kappa(-), \mathcal{S}et)) \ .$$

The construction of $\kappa \copyright_{\mathbf{E}} \mu$ can be described as follows. For all point E of $\mathbf{E}$, and all $x \in \mu(E)$, take a copy $\kappa(E,x)$ of the weft $\kappa(E)$: for example two copies $\kappa(\mathtt{Pt}, U)$ and $\kappa(\mathtt{Pt}, N)$ of $\kappa(\mathtt{Pt})$, five copies $\kappa(\mathtt{Ar}, x)$ of $\kappa(\mathtt{Ar})$, for $x \in \{z, s, p, id_N, p \circ s\}$, *etc.* For all arrow $e : E \to E'$ of $\mathbf{E}$, and all $x \in \mu(E)$, take a copy $\kappa(e,x) : \kappa(E', x') \to \kappa(E, x)$, where $x' = \mu(e)(x) \in \mu(E')$, of the homomorphism $\kappa(e) : \kappa(E') \to \kappa(E)$: for example a copy $\kappa(\mathtt{dom}, z) : \kappa(\mathtt{Pt}, U) \to \kappa(\mathtt{Ar}, z)$ of $\kappa(\mathtt{dom})$, *etc.* Then, the wefts $\kappa(E,x)$ are merged together according to the homomorphisms $\kappa(e,x)$: for example the line $\kappa(\mathtt{Pt})_1$ of the weft $\kappa(\mathtt{Ar}, z)$ is merged with $\kappa(\mathtt{Pt}, U)$, thanks to $\kappa(\mathtt{dom}, z)$.

In this way we *amplify* each ingredient x of μ of nature E into $\kappa(E)$, and we merge them all together in the most natural way, *i.e.* in the same way as the ingredients x of μ are merged together for building μ. Such a construction is an *inductive limit* in the category of wefts of ambigraphs (a short survey about projective and inductive limits can be found in the appendix of [guide1]).

Although we have not yet taken into account the constraints of $\mathbf{S}$, it should be noted that there are four constraints in $\kappa \copyright_{\mathbf{E}} \mu$, arising from the constraints of the wefts $\kappa(-)$:

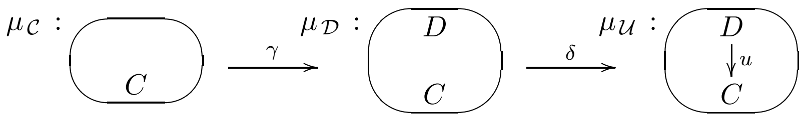
$$H'_U = H_U + H^e_U \ , \ H^e_U = \mathbb{1} \ , \ H'_N = H_N + H^e_N \ , \ H^e_N = \mathbb{1} \ .$$

2.8 Terminal point constraint

Let us now consider the constraint "$U = \mathbb{1}$" of $\mathbf{S}$, which means that the interpretation of U should be a terminal point. Let:

$$\mathbf{S}_1 : \mathbf{E} \stackrel{\frown}{\longrightarrow} \mathcal{S}et$$

be the weft of ambigraphs of support μ and constraint Γ, where Γ, which is denoted Γ_T in [guide1, section 2.2], is the constraint over μ of level 0, with potential:



and body $\chi : \mu_{\mathcal{C}} \to \mu$ such that $\chi(C) = U$.

Both points of view used in 2.7 for the support can be extended to the constraint.

1. The first point of view considers the realizations of $\mathbf{S}_1$ towards $\Theta \circ \kappa$, *i.e.* the elements of the set:

$$\boxed{\mathrm{Real}_{\mathcal{M}od(\mathbf{E})}(\mathbf{S}_1, \Theta \circ \kappa) \ .}$$

They are the ambifunctors from μ to $\Theta \circ \kappa$ which satisfy the constraint Γ. This means that the interpretation of U by such a realization, denoted $X' = X \sqcup \{\varepsilon_X\}$, satisfies the following property: for all set $Y' = Y \sqcup \{\varepsilon_Y\}$, there is a unique map $f' : Y' \to X'$, such that $f'(\varepsilon_Y) = \varepsilon_X$. Equivalently, the interpretation of U is, among the sets with an error, terminal with respect to the maps which propagate the error. It is easy to check that X' is a one-element set, *i.e.* that X is empty. But this does *not* meet our requirement: according to 2.1, the interpretation of U should be the two-element set $\mathbb{U}' = \{*, \varepsilon_{\mathbb{U}}\}$.

2. Similarly to the construction of $\kappa \copyright_{\mathbf{E}} \mu$ in 2.7, we may build the three wefts of ambigraphs $\kappa \copyright_{\mathbf{E}} \mu_{\mathcal{C}}$, $\kappa \copyright_{\mathbf{E}} \mu_{\mathcal{D}}$ and $\kappa \copyright_{\mathbf{E}} \mu_{\mathcal{U}}$, with the extension homomorphisms $\kappa \copyright_{\mathbf{E}} \gamma$ and $\kappa \copyright_{\mathbf{E}} \delta$. Denoting $\copyright$ for $\copyright_{\mathbf{E}}$ and representing each line $\kappa(\mathtt{Pt})_n$ as:

$$\bullet \longrightarrow + \longleftarrow \mathbb{1}$$

the potential we get is the following:

$$\kappa \copyright \mu_{\mathcal{C}} : \quad \left(\bullet \rightarrow + \leftarrow \mathbb{1} \right) \qquad \kappa \copyright \mu_{\mathcal{D}} : \quad \left(\begin{array}{c} \bullet \rightarrow + \leftarrow \mathbb{1} \\ \bullet \rightarrow + \leftarrow \mathbb{1} \end{array} \right) \qquad \kappa \copyright \mu_{\mathcal{U}} : \quad \left(\begin{array}{ccccccc} \bullet & \rightarrow & + & \leftarrow & \mathbb{1} \\ & & \downarrow & \equiv & \downarrow \\ \bullet & \rightarrow & + & \leftarrow & \mathbb{1} \end{array} \right)$$

$$\xrightarrow{\kappa \copyright \gamma} \qquad \xrightarrow{\kappa \copyright \delta}$$

Let us denote $\langle \kappa \copyright_{\mathbf{E}} \chi \rangle$ (this notation will become clear in 3.2) the ambifunctor from $Supp(\kappa \copyright_{\mathbf{E}} \mu_{\mathcal{C}})$ towards $Supp(\kappa \copyright_{\mathbf{E}} \mu)$ defined by:

$$(H_C \xrightarrow{h_C} H'_C \xleftarrow{h^e_C} H^e_C) \quad \mapsto \quad (H_U \xrightarrow{h_U} H'_U \xleftarrow{h^e_U} H^e_U) \; .$$

In this way we get a constraint $\langle \kappa \copyright_{\mathbf{E}} \Gamma \rangle$ over the ambigraph support of $\kappa \copyright_{\mathbf{E}} \mu$. Let $\langle \kappa \copyright_{\mathbf{E}} \mathbf{S}_1 \rangle$ denote the weft of ambigraphs made of the support of $\kappa \copyright_{\mathbf{E}} \mu$, the four constraints of $\kappa \copyright_{\mathbf{E}} \mu$, and the constraint $\langle \kappa \copyright_{\mathbf{E}} \Gamma \rangle$ (over the support of $\kappa \copyright_{\mathbf{E}} \mu$, also). The elements of the set:

$$\boxed{\Theta(\langle \kappa \copyright_{\mathbf{E}} \mathbf{S}_1 \rangle) \; .}$$

are the set-valued realizations of $\langle \kappa \copyright_{\mathbf{E}} \mathbf{S}_1 \rangle$, *i.e.* the set-valued realizations of $\kappa \copyright_{\mathbf{E}} \mu$ which satisfy the constraint $\langle \kappa \copyright_{\mathbf{E}} \Gamma \rangle$. It is easy to check that in this way, as above, the interpretation of U should be a one-element set, and not a two-element set as required.

In this example, neither of these two points of view meets our requirements. However both are equivalent. Indeed corollary 1 will prove the existence of a canonical bijection:

$$\boxed{\Theta(\langle \kappa \copyright_{\mathbf{E}} \mathbf{S}_1 \rangle) \cong \mathrm{Real}_{\mathcal{M}od(\mathbf{E})}(\mathbf{S}_1, \Theta \circ \kappa) \; ,}$$

or, equivalently:

$$\mathrm{Real}_{\mathcal{A}mbi}(\langle \kappa \copyright_{\mathbf{E}_{\mathcal{A}mbi}} \mathbf{S}_1 \rangle, \mathcal{S}et) \cong \mathrm{Real}_{\mathcal{A}mbi}(\mathbf{S}_1, \mathrm{Real}_{\mathcal{A}mbi}(\kappa(-), \mathcal{S}et)) \; .$$

2.9 Initiality constraint

Now let us handle in the same two ways the initiality constraint of $\mathbf{S}$, which says that the interpretation of $(U \xrightarrow{z} N \xrightarrow{s} N)$ is initial among the $(X_0 \xrightarrow{x_0} X \xrightarrow{f} X)$, where X_0 is terminal. It is easy to check that we get the following result: the interpretation of N is (up to isomorphism) the same as the interpretation of U, *i.e.* a one-element set, and the interpretation of z, as the interpretation of s, is the identity. This means that natural numbers have disapppeared... this is not at all what is required!

2.10 Conclusion

The previous study is related to the approach using monads [Moggi 91, Wadler 85]: indeed the ambigraph $\Theta \circ \kappa$ is related to the Kleisli category of the monad $M(X) = 1 + X$. Note that a map $f' : X'_1 \to X'_2$ which propagates the error is characterized by its restriction $f'|_{X_1} : X_1 \to X_2 \sqcup \{\varepsilon_{X_2}\}$. This means that the set-valued realizations of $\kappa(\mathtt{Ar})$ can be identified to the set-valued realizations of the following weft of ambigraphs:

$$\begin{tikzcd}
H_1 \arrow[r, "h_1"] \arrow[dr] & H'_1 = H_1 + H^e_1 & H^e_1 = \mathbb{1} \arrow[l, "h^e_1"'] \\
H_2 \arrow[r, "h_2"] & H'_2 = H_2 + H^e_2 & H^e_2 = \mathbb{1} \arrow[l, "h^e_2"']
\end{tikzcd}$$

To sum up, this first way to deal with the example of natural numbers with predecessor is not entirely satisfactory:

(A) the support of $\mathbf{S}$ behaves fairly well: we have formalized the fact that in each set there is an error element, and that each map propagates the error;

(B) however, even on the support of $\mathbf{S}$, we have not been able to say that the interpretations of z and s should not create an error, while the interpretation of $p \circ z$ should always return an error;

(C) and what is not satisfactory at all is that our formalism is not able to deal properly with the constraints of $\mathbf{S}$.

Our approach relies on the idea of giving additional information on the required interpretation of each ingredient of $\mathbf{S}$: for instance, each map should propagate the error. Point (B) above will be solved in 5 by giving more accurate information: each map should propagate the error, and in addition some maps should not create an error, whereas some others should always return an error. For this purpose, we will build a *stratification* and a *ribbon product*, as defined in 4. In 5 we will see that this does indeed solve point (B), and that, in addition, it solves point (C).

Finally, let us see what happens when dealing with a binary product constraint.

2.11 Product constraint

Let us consider the cartesian product $\mathbb{N}^2 = \mathbb{N} \times \mathbb{N}$ and the addition $+ : \mathbb{N}^2 \to \mathbb{N}$. Since any attempt to compute the predecessor of 0 should return an error, the addition should return an error as soon as one of its arguments is an error. Going on with the analysis made in 2.1, this means that we need a new error element $\varepsilon_{\mathbb{N}^2}$, the set $(\mathbb{N}^2)' = \mathbb{N}^2 \sqcup \{\varepsilon_{\mathbb{N}^2}\}$ and the map $+' : (\mathbb{N}^2)' \to \mathbb{N}'$ which extends the addition and propagates the error.

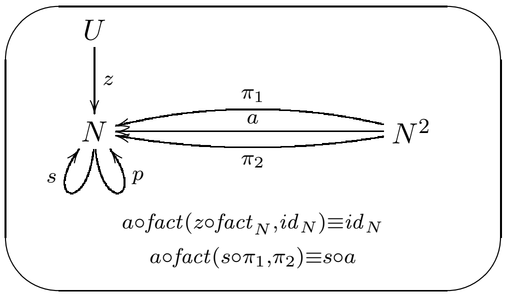
Let us replace the weft $\mathbf{S}$ by the following one (since the constraints of $\mathbf{S}$ do not behave well, we forget about them):

$\mathcal{A}mbi$-WEFT $\mathbf{S}_+$:

extends:	$\underline{\mathbf{S}}$,
point:	N^2 ,
arrows:	$\pi_1 : N^2 \to N$, $\pi_2 : N^2 \to N$, $a : N^2 \to N$,
equations:	$a \circ \mathit{fact}(z \circ \mathit{fact}_N, id_N) \equiv id_N$, $a \circ \mathit{fact}(s \circ \pi_1, \pi_2) \equiv s \circ a$,
product:	$N \xleftarrow{\pi_1} N^2 \xrightarrow{\pi_2} N$.

The support of $\mathbf{S}_+$ is:

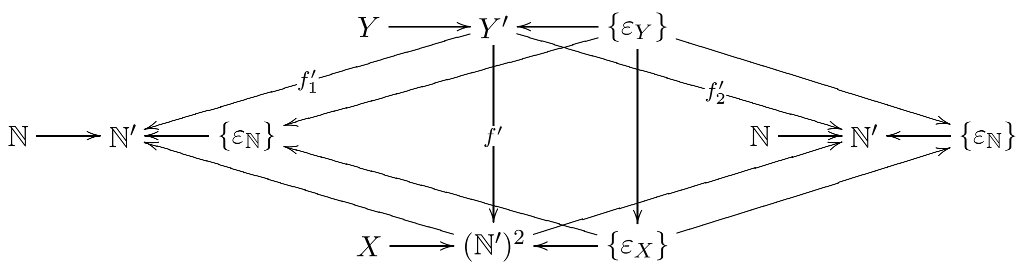
$\underline{\mathbf{S}}_+$:



We consider a realization of $\mathbf{S}_+$ towards $\Theta \circ \kappa$ which interprets U, N, z, s and p as $\mathbb{U}'$, $\mathbb{N}'$, $0'$, succ' and pred' respectively, and we look at the way it interprets N^2, π_1 and π_2.

Let $\varepsilon_X = (\varepsilon_{\mathbb{N}}, \varepsilon_{\mathbb{N}}) \in (\mathbb{N}')^2$, and let X be the complement of $\{\varepsilon_X\}$ in $(\mathbb{N}')^2$. Then $(\mathbb{N}')^2 = X \sqcup \{\varepsilon_X\}$ is a point of $\Theta \circ \kappa$, since it is a set-valued realization of $\kappa(\mathtt{Pt})$. Both projections p_1 and p_2 of $(\mathbb{N}')^2$ on $\mathbb{N}'$ are arrows of $\Theta \circ \kappa$, since they are set-valued realizations of $\kappa(\mathtt{Ar})$. We now prove that $\mathbb{N}' \xleftarrow{p_1} (\mathbb{N}')^2 \xrightarrow{p_2} \mathbb{N}'$ is the interpretation of $N \xleftarrow{\pi_1} N^2 \xrightarrow{\pi_2} N$.

First, in the category of sets, since $\mathbb{N}' \xleftarrow{p_1} (\mathbb{N}')^2 \xrightarrow{p_2} \mathbb{N}'$ is a cartesian product, for all set Y' and all pair of maps $(f'_1 : Y' \to \mathbb{N}', f'_2 : Y' \to \mathbb{N}')$, the map $f' = \mathit{fact}(f'_1, f'_2) : Y' \to (\mathbb{N}')^2$ is defined by $f'(y') = (f'_1(y'), f'_2(y'))$, for all $y' \in Y'$. If in addition Y' has an error element and if f'_1 and f'_2 propagate the error, then f' will also propagate the error. It proves that $\mathbb{N}' \xleftarrow{p_1} (\mathbb{N}')^2 \xrightarrow{p_2} \mathbb{N}'$ is a product in $\Theta \circ \kappa$.



In this way (up to isomorphism) N^2, π_1 and π_2 are interpreted as $(\mathbb{N}')^2$, p_1 and p_2. But this is not was we wish. Indeed, $(\mathbb{N}')^2 = (\mathbb{N} \sqcup \{\varepsilon_{\mathbb{N}}\})^2$ is different from $(\mathbb{N}^2)' = \mathbb{N}^2 \sqcup \{\varepsilon_{\mathbb{N}^2}\}$: it has elements like $(\varepsilon_{\mathbb{N}}, x_2)$ with $x_2 \neq \varepsilon_{\mathbb{N}}$, and $(x_1, \varepsilon_{\mathbb{N}})$ with $x_1 \neq \varepsilon_{\mathbb{N}}$, which we want to avoid.

It follows that of these three constraints (in 2.8, 2.9 and 2.11), none is correctly handled by the crown product. We will see in 5 that they can be handled correctly thanks to a ribbon product. However, in 3, we will use this section in order to illustrate the definition of the crown product.

3 Crown product

This section is devoted to the study of the *crown product.* Let:

- $\mathbf{E}$ be a projective sketch,
- $\mathcal{W}$ a *cocomplete* category, *i.e.* a category such that for all base $\eta : \mathcal{B} \to \mathcal{W}$ there is in $\mathcal{W}$ an inductive limit cone with base η,
- $\kappa : \mathbf{E} \mathrel{+\!\!\!\longrightarrow} \mathcal{W}$ a counter-model of $\mathbf{E}$,
- and $\Theta : \mathcal{W} \mathrel{+\!\!\!\longrightarrow} \mathcal{S}et$ a contravariant functor which is *left-exact*, *i.e.* which maps each inductive limit cone in $\mathcal{W}$ to a projective limit cone in $\mathcal{S}et$.

$$\begin{tikzcd} & \mathbf{E} \arrow[dl, "\kappa"', "\shortmid" marking] \\ \mathcal{W} \arrow[dr, "\Theta"', "\shortmid" marking] & \\ & \mathcal{S}et \end{tikzcd}$$

The crown product $\kappa \copyright_{\mathbf{E}} \mu$ of κ by a model μ of $\mathbf{E}$ is defined in 3.1. Proposition 1 generalizes 2.7: $\kappa \copyright_{\mathbf{E}} \mu$ is a point of $\mathcal{W}$ which is mapped by Θ on the set of homomorphisms from μ towards $\Theta \circ \kappa$.

Then in 3.2 we use the functoriality of $\kappa \copyright_{\mathbf{E}} \mu$ with respect to μ in order to define the crown product $\kappa \copyright_{\mathbf{E}} \mathbf{S}$ of κ by a weft $\mathbf{S}$ of models of $\mathbf{E}$. The fundamental result on the crown product is theorem 1.

In 3.3 we assume that $\mathcal{W} = \mathcal{W}\!\mathit{eft}(\mathcal{A})$. In this important case we prove corollary 1. This result generalizes what could be seen in 2.8: $\langle \kappa \copyright_{\mathbf{E}} \mathbf{S} \rangle$ is a point of $\mathcal{W}$, and it is mapped by $\mathrm{Real}_{\mathcal{A}}(-, A)$, for all point A of $\mathcal{A}$, on the set of realizations of $\mathbf{S}$ towards $\mathrm{Real}_{\mathcal{A}}(\kappa(-), A)$.

These definitions and results can be illustrated by the example considered in 2.

3.1 Support

We may now define the *crown product* $\kappa \copyright_{\mathbf{E}} \mu$ of κ by a model μ of $\mathbf{E}$. It is a point of $\mathcal{W}$, and its image by Θ is characterized by proposition 1.

Definition 1 Let μ be a model of $\mathbf{E}$. The *crown product of* κ *by* μ *over* $\mathbf{E}$ is the point of $\mathcal{W}$:

$$\kappa \copyright_{\mathbf{E}} \mu = \mathit{IndLim}(\underline{\mathbf{E} \backslash \mu} \xrightarrow{\mathbf{E} \backslash\!\backslash \mu} \underline{\mathbf{E}} \overset{\kappa}{\mathrel{+\!\!\!\longrightarrow}} \mathcal{W}) \ .$$

In this definition, though $\kappa \circ (\mathbf{E} \backslash\!\backslash \mu)$ is a model of $\mathbf{E} \backslash \mu$ towards $\mathcal{W}$, it is only the sublying functor which is used for the construction of the inductive limit. We may also write:

$$\kappa \copyright_{\mathbf{E}} \mu = \mathit{IndLim}_{\{[E,x]\}}(\kappa(E)) \, ,$$

where $[E, x]$ is a point of $\mathbf{E} \backslash \mu$ (which means that E is a point of $\mathbf{E}$ and x an element of $\mu(E)$).

This construction runs as for the example in 2: it *amplifies* each ingredient x of μ of nature E into $\kappa(E)$, and merges these amplifications. More precisely, for all point $[E, x]$ of $\mathbf{E} \backslash \mu$, we consider a copy $\kappa(E, x)$ of the point $\kappa(E)$ of $\mathcal{W}$. For all arrow $[e, x_1] : [E_1, x_1] \to [E_2, x_2]$ (where $x_2 = \mu(e)(x_1)$) of $\mathbf{E} \backslash \mu$ we consider a copy $\kappa(e, x_1) : \kappa(E_2, x_2) \to \kappa(E_1, x_1)$ of the arrow $\kappa(e) : \kappa(E_2) \to \kappa(E_1)$ of $\mathcal{W}$. Then, in the category $\mathcal{W}$, these points $\kappa(E, x)$ are merged together

according to the arrows $\kappa(e,x)$: this is the inductive limit $IndLim_{\{[E,x]\}}(\kappa(E))$, which does exist, because $\mathcal{W}$ is cocomplete.

Let $h:\mu_1 \to \mu_2$ be a homomorphism of models of $\mathbf{E}$. We define the arrow of $\mathcal{W}$:

$$\kappa \copyright_{\mathbf{E}} h : \kappa \copyright_{\mathbf{E}} \mu_1 \to \kappa \copyright_{\mathbf{E}} \mu_2$$

from the identity arrows from $\kappa(E,x_1)$ towards $\kappa(E,h(E)(x_1))$ (both are equal to $\kappa(E)$) for all point E of $\mathbf{E}$ and all ingredient x_1 of μ_1 of nature E.

In this way we get the *crown product functor*:

$$\kappa \copyright_{\mathbf{E}} - : \mathcal{M}od(\mathbf{E}) \longrightarrow \mathcal{W} \ .$$

The fundamental property of the crown product, below, states that there is a canonical bijection between the set $\Theta(\kappa \copyright_{\mathbf{E}} \mu)$ and the set of homomorphisms of μ towards $\Theta \circ \kappa$ (as models of $\mathbf{E}$).

Proposition 1 *For all model μ of $\mathbf{E}$ there is a canonical bijection:*

$$\boxed{\Theta(\kappa \copyright_{\mathbf{E}} \mu) \cong \mathrm{Hom}_{\mathcal{M}od(\mathbf{E})}(\mu, \Theta \circ \kappa) \ .}$$

$$\begin{tikzcd}
 & \mathbf{E} \arrow[dl, "\kappa"'] \arrow[dd, "\mu"] \\
\kappa \copyright_{\mathbf{E}} \mu \in \mathcal{W} \arrow[dr, "\Theta"'] & \\
 & \mathcal{S}et
\end{tikzcd}$$

About the proof.
The proof of this result will be given in our reference manual. It makes use of the Yoneda lemma for projective sketches, from [guide1, section 4.4]. Here we only give an idea of the proof.

First let us consider the lefthand side $\Theta(\kappa \copyright_{\mathbf{E}} \mu)$. Since $\kappa \copyright_{\mathbf{E}} \mu$ is the inductive limit of the points $\kappa(E,x)$ and since Θ is left-exact, $\Theta(\kappa \copyright_{\mathbf{E}} \mu)$ is the projective limit of the sets $\Theta(\kappa(E,x))$: an element y of $\Theta(\kappa \copyright_{\mathbf{E}} \mu)$ is a compatible family of elements of $\Theta(\kappa(E,x))$. It means that $y=(y_{[E,x]})_{\{[E,x]\}}$ with each $y_{[E,x]} \in \Theta(\kappa(E,x))$, and for all $e:E \to E'$, with $x'=\mu(e)(x)$, the following compatibility condition holds: $\Theta(\kappa(e,x)) : y_{[E,x]} \mapsto y_{[E',x']}$.

Now let us consider the righthand side $\mathrm{Hom}_{\mathcal{M}od(\mathbf{E})}(\mu, \Theta \circ \kappa)$. The model μ of $\mathbf{E}$ can be seen as the inductive limit of all its ingredients, arranged according to their nature (this is the Yoneda lemma for projective sketches). Then a homomorphism h of models of $\mathbf{E}$ with domain μ can be described as the compatible family of images of all the ingredients of μ. Let $y_{[E,x]} = h(x)$ for all point E of $\mathbf{E}$ and all $x \in \mu(E)$, then h is characterized by the family $y=(y_{[E,x]})_{\{[E,x]\}}$. By definition of a homomorphism the following compatibility condition holds: $\Theta(\kappa(e,x)) : y_{[E,x]} \mapsto y_{[E',x']}$.

Finally, we have given the same description for both hand sides: the theorem follows.
$\Diamond$

When $\mathcal{W}=\mathcal{M}od(\mathbf{E})$, which is a cocomplete category, we may consider the Yoneda counter-model (see [guide1, section 4.4]):

$$\mathcal{Y}_{\mathbf{E}} : \mathbf{E} \mathrel{+\!\!\!\longrightarrow} \mathcal{M}od(\mathbf{E}) \ .$$

Given the definition of the crown product, the Yoneda lemma for projective sketches states that:

$$\mathcal{Y}_{\mathbf{E}} \copyright_{\mathbf{E}} \mu = \mu \ .$$

It means that the Yoneda counter-model is a *unit* for the crown product.

The definition of the crown product uses a blow-up of the composition graph $\underline{\mathbf{E}}$, in order to count the ingredients of $\mathbf{E}$ in a proper way. However it does not use the fundamental result on the blow-up, as stated in [guide1, section 4.2].

3.2 Constraints

We now define the *crown product* $\kappa \circledC_{\mathbf{E}} \mathbf{S}$ of κ by a $\mathcal{M}od(\mathbf{E})$-weft $\mathbf{S}$. This is a $\mathcal{W}$-weft, and its definition is quite simple, thanks to the functoriality of the crown product $\kappa \circledC_{\mathbf{E}} \mu$ with respect to μ. Theorem 1 is easily deduced from proposition 1. It characterizes the image of $\kappa \circledC_{\mathbf{E}} \mathbf{S}$ by a functor $\Theta_{\mathcal{W}eft}$ which is defined from Θ.

As noted in [guide1, section 2.6], the functor $\kappa \circledC_{\mathbf{E}} - : \mathcal{M}od(\mathbf{E}) \to \mathcal{W}$ can be extended to a functor $\mathcal{W}eft(\kappa \circledC_{\mathbf{E}} -)$ (which is still denoted $\kappa \circledC_{\mathbf{E}} -$, since this cannot lead to any mistake) between the corresponding categories of wefts:

$$\kappa \circledC_{\mathbf{E}} - = \mathcal{W}eft(\kappa \circledC_{\mathbf{E}} -) \ : \ \mathcal{W}eft(\mathcal{M}od(\mathbf{E})) \to \mathcal{W}eft(\mathcal{W}) \ .$$

This functor $\kappa \circledC_{\mathbf{E}} -$ has been defined recursively, according to the level of the wefts. For all $\mathcal{M}od(\mathbf{E})$-weft $\mathbf{S}$, the $\mathcal{W}$-weft $\kappa \circledC_{\mathbf{E}} \mathbf{S}$ has for support $\kappa \circledC_{\mathbf{E}} \underline{\mathbf{S}}$ and for constraints $\kappa \circledC_{\mathbf{E}} \Gamma$ for all the constraints Γ of $\mathbf{S}$.

Definition 2 Let $\mathbf{S}$ be a weft of models of $\mathbf{E}$. The *crown product of* κ *by* $\mathbf{S}$ *over* $\mathbf{E}$ is the $\mathcal{W}$-weft:

$$\kappa \circledC_{\mathbf{E}} \mathbf{S} \ .$$

On the other hand, for all point M of $\mathcal{W}$, the definition of the realizations of a weft, as given in [guide1, section 2.5], gives rise to an extension of the contravariant functor:

$$\mathrm{Hom}_{\mathcal{W}}(-, M) : \mathcal{W} \longrightarrow \mathcal{S}et$$

to a contravariant functor:

$$\mathrm{Real}_{\mathcal{W}}(-, M) : \mathcal{W}eft(\mathcal{W}) \longrightarrow \mathcal{S}et \ .$$

This is generalized by the definition, given below, of the extension:

$$\Theta_{\mathcal{W}eft} : \mathcal{W}eft(\mathcal{W}) \longrightarrow \mathcal{S}et$$

of the contravariant functor Θ. Indeed, whenever $\Theta = \mathrm{Hom}_{\mathcal{W}}(-, M)$, we will get:

$$(\mathrm{Hom}_{\mathcal{W}}(-, M))_{\mathcal{W}eft} = \mathrm{Real}_{\mathcal{W}}(-, M) \ .$$

Definition 3 For all $\mathcal{W}$-weft $\mathbf{T}$, the set $\Theta_{\mathcal{W}eft}(\mathbf{T})$ will be a subset of $\Theta(\underline{\mathbf{T}})$, and for all homomorphism $\sigma : \mathbf{T} \to \mathbf{T}'$ of $\mathcal{W}$-wefts, the map $\Theta_{\mathcal{W}eft}(\sigma) : \Theta_{\mathcal{W}eft}(\mathbf{T}') \to \Theta_{\mathcal{W}eft}(\mathbf{T})$ will be the restriction of $\Theta(\underline{\sigma})$.

If $\mathbf{T}$ is a $\mathcal{W}$-weft of level 0, then $\Theta_{\mathcal{W}eft}$ is Θ.

Now let n be an integer ≥ 0. Assume that $\Theta_{\mathcal{W}eft}$ is defined on the $\mathcal{W}$-wefts of level $\leq n-1$. Let $\mathbf{T}$ be a $\mathcal{W}$-weft of level n, and let $\Gamma = (\underline{\mathbf{C}} \xrightarrow{\chi} \underline{\mathbf{T}}, \mathbf{C} \xrightarrow{\gamma} \mathbf{D} \xrightarrow{\delta} \mathbf{U})$ be a constraint of $\mathbf{T}$. Then an element x of $\Theta(\underline{\mathbf{T}})$ *satisfies* the constraint Γ if:

- the element $x_{\mathbf{C}} = \Theta(\chi)(x)$ of $\Theta(\underline{\mathbf{C}})$ is in $\Theta_{\mathcal{W}eft}(\mathbf{C})$;

- and for all element $x_{\mathbf{D}}$ of $\Theta_{\mathcal{W}eft}(\mathbf{D})$ such that $\Theta(\gamma)(x_{\mathbf{D}})=x_{\mathbf{C}}$, there is a unique element $x_{\mathbf{U}}$ of $\Theta_{\mathcal{W}eft}(\mathbf{U})$ such that $\Theta(\delta)(x_{\mathbf{U}})=x_{\mathbf{D}}$.

$$\begin{tikzcd}
\underline{\mathbf{C}} \arrow[r, hook] \arrow[d, "\chi"'] & \mathbf{C} \arrow[r, "\gamma"] & \mathbf{D} \arrow[r, "\delta"] & \mathbf{U} \\
\underline{\mathbf{T}} & & &
\end{tikzcd}$$

$$\begin{tikzcd}
\Theta(\underline{\mathbf{C}}) & \Theta_{\mathcal{W}eft}(\mathbf{C}) \arrow[l, hook'] & \Theta_{\mathcal{W}eft}(\mathbf{D}) \arrow[l, "\Theta_{\mathcal{W}eft}(\gamma)"'] & \Theta_{\mathcal{W}eft}(\mathbf{U}) \arrow[l, "\Theta_{\mathcal{W}eft}(\delta)"'] \\
\Theta(\underline{\mathbf{T}}) \arrow[u, "\Theta(\chi)"] & & &
\end{tikzcd}$$

Then $\Theta_{\mathcal{W}eft}(\mathbf{T})$ is the set of the elements of $\Theta(\underline{\mathbf{T}})$ which satisfy all the constraints of $\mathbf{T}$.

The fundamental property of the crown product of κ by a weft, below, states that there is a canonical bijection between the set $\Theta_{\mathcal{W}eft}(\kappa \circledC_{\mathbf{E}} \mathbf{S})$ and the set of realizations of $\mathbf{S}$ towards $\Theta \circ \kappa$.

Theorem 1 (Fundamental property of the crown product) *For all $\mathcal{M}od(\mathbf{E})$-weft $\mathbf{S}$ there is a canonical bijection:*

$$\boxed{\Theta_{\mathcal{W}eft}(\kappa \circledC_{\mathbf{E}} \mathbf{S}) \cong \mathrm{Real}_{\mathcal{M}od(\mathbf{E})}(\mathbf{S}, \Theta \circ \kappa)\ .}$$

$$\begin{tikzcd}
 & & \mathbf{E} \arrow[dl, "\kappa"'] \arrow[dd, "\mathbf{S}"] \\
\kappa \circledC_{\mathbf{E}} \mathbf{S} \in \mathcal{W}eft(\mathcal{W}) \arrow[drr, bend right, "\Theta_{\mathcal{W}eft}"'] & \mathcal{W} \arrow[l, hook'] \arrow[dr, "\Theta"] \arrow[d, phantom, "="] & \\
 & & \mathcal{S}et
\end{tikzcd}$$

About the proof. The proof proceeds recursively, according to the level, and uses proposition 1: see the reference manual.
◊

3.3 Wefts of wefts

Now let us focus on the special case where $\mathcal{W}$ itself is a category of wefts:

$$\mathcal{W} = \mathcal{W}eft(\mathcal{A})$$

for some cocomplete category $\mathcal{A}$ (so that $\mathcal{W}$ is cocomplete), and where:

$$\Theta = \mathrm{Real}_{\mathcal{A}}(-, A)$$

for some point A of $\mathcal{A}$: this is the case in our example (in 2) with $\mathcal{A}=\mathcal{A}mbi$ and $A=\mathcal{S}et$. Then $\kappa \circledC_{\mathbf{E}} \mathbf{S}$ is a weft of $\mathcal{A}$-wefts. We will assign to it a $\mathcal{A}$-weft $\langle \kappa \circledC_{\mathbf{E}} \mathbf{S} \rangle$ in such a way that theorem 1 implies a characterisation of the image of $\langle \kappa \circledC_{\mathbf{E}} \mathbf{S} \rangle$ by the functor $\mathrm{Real}_{\mathcal{A}}(-, A)$.

For all category $\mathcal{A}$, we will build a functor

$$\langle - \rangle : \mathcal{W}eft(\mathcal{W}eft(\mathcal{A})) \to \mathcal{W}eft(\mathcal{A})$$

such that for all $\mathcal{W}\!\mathit{eft}(\mathcal{A})$-weft $\mathbf{S}$:

$$\underline{\langle \mathbf{S} \rangle} = \underline{\underline{\mathbf{S}}} \qquad (\text{in } \mathcal{A}) ,$$

and for all point A of $\mathcal{A}$ there is a canonical bijection:

$$\Theta_{\mathcal{W}\!\mathit{eft}}(\mathbf{S}) = (\mathrm{Real}_{\mathcal{A}}(\mathbf{S}, A))_{\mathcal{W}\!\mathit{eft}} \cong \mathrm{Real}_{\mathcal{A}}(\langle \mathbf{S} \rangle, A) .$$

This property states that the "generalized" realizations from $\mathbf{S}$ towards A can be identified to the realizations of $\langle \mathbf{S} \rangle$ towards A.

$$\begin{tikzcd}
\mathcal{W}\!\mathit{eft}(\mathcal{W}\!\mathit{eft}(\mathcal{A})) \arrow[r, "\langle - \rangle"] \arrow[rr, bend right, "(\mathrm{Real}_{\mathcal{A}}(-,A))_{\mathcal{W}\!\mathit{eft}}"', "/" marking] & \mathcal{W}\!\mathit{eft}(\mathcal{A}) \arrow[r, "\mathrm{Real}_{\mathcal{A}}(-,A)", "/" marking] \arrow[d, phantom, "\cong"] & \mathcal{S}\!\mathit{et} \\
 & {} &
\end{tikzcd}$$

The functor $\langle - \rangle : \mathcal{W}\!\mathit{eft}(\mathcal{W}) \to \mathcal{W}\!\mathit{eft}(\mathcal{A})$ may now be defined recursively, according to the level of the $\mathcal{W}\!\mathit{eft}(\mathcal{W})$-weft $\mathbf{S}$.

Definition 4 If $\mathbf{S}$ has level 0, then $\langle \mathbf{S} \rangle$ is its support $\underline{\mathbf{S}}$.
If $\sigma : \mathbf{S} \to \mathbf{S}'$ has level 0, then $\langle \sigma \rangle = \underline{\sigma}$.

If $\mathbf{S}$ has level $n \geq 1$ then $\langle \mathbf{S} \rangle$ is made of the $\mathcal{A}$-weft $\underline{\mathbf{S}}$ together with, for all constraint Γ of $\mathbf{S}$, the constraint $\langle \Gamma \rangle$ over $\underline{\underline{\mathbf{S}}}$ defined as follows, using the functoriality of $\langle - \rangle$. Let us consider the body of Γ:

$$\chi : \underline{\mathbf{C}} \to \underline{\mathbf{S}} \qquad (\text{ in } \mathcal{W})$$

and its potential:

$$\mathbf{C} \xrightarrow{\gamma} \mathbf{D} \xrightarrow{\delta} \mathbf{U} \qquad (\text{ in } \mathcal{W}\!\mathit{eft}(\mathcal{W})) .$$

Then $\langle \Gamma \rangle$ is the constraint with body:

$$\underline{\chi} : \underline{\underline{\mathbf{C}}} \to \underline{\underline{\mathbf{S}}} \qquad (\text{ in } \mathcal{A})$$

and with potential:

$$\langle \mathbf{C} \rangle \xrightarrow{\langle \gamma \rangle} \langle \mathbf{D} \rangle \xrightarrow{\langle \delta \rangle} \langle \mathbf{U} \rangle \qquad (\text{ in } \mathcal{W}\!\mathit{eft}(\mathcal{A})) .$$

This is indeed a constraint over $\underline{\underline{\mathbf{S}}}$, because $\underline{\langle \mathbf{C} \rangle} = \underline{\underline{\mathbf{C}}}$.
If $\sigma : \mathbf{S} \to \mathbf{S}'$ has level $n \geq 1$, then we check $\underline{\underline{\sigma}} : \underline{\underline{\mathbf{S}}} \to \underline{\underline{\mathbf{S}'}}$ defines a homomorphism of $\mathcal{A}$-wefts, which is denoted:

$$\langle \sigma \rangle : \langle \mathbf{S} \rangle \to \langle \mathbf{S}' \rangle .$$

Then clearly $\underline{\langle \mathbf{S} \rangle} = \underline{\underline{\mathbf{S}}}$, and it is easy to check that $(\mathrm{Real}_{\mathcal{A}}(-, A))_{\mathcal{W}\!\mathit{eft}}(\mathbf{S}) \cong \mathrm{Real}_{\mathcal{A}}(\langle \mathbf{S} \rangle, A)$ for all point A of $\mathcal{A}$.

The following result follows immediately from theorem 1:

Corollary 1 *Assume that $\mathcal{W} = \mathcal{W}\!\mathit{eft}(\mathcal{A})$ for some cocomplete category $\mathcal{A}$. For all point A of $\mathcal{A}$ and all $\mathcal{M}\!\mathit{od}(\mathbf{E})$-weft $\mathbf{S}$ there is a canonical bijection:*

$$\boxed{\mathrm{Real}_{\mathcal{A}}(\langle \kappa \copyright_{\mathbf{E}} \mathbf{S} \rangle, A) \cong \mathrm{Real}_{\mathcal{M}\!\mathit{od}(\mathbf{E})}(\mathbf{S}, \mathrm{Real}_{\mathcal{A}}(\kappa(-), A)) .}$$

$$\begin{tikzcd}
 & \mathbf{E} \arrow[dl, "\kappa"', "\backslash" marking] \arrow[d, "\mathbf{S}", ")" marking] \\
\langle \kappa \copyright_{\mathbf{E}} \mathbf{S} \rangle \in \mathcal{W}\!\mathit{eft}(\mathcal{A}) = \mathcal{W} \arrow[r, "\mathrm{Real}_{\mathcal{A}}(-,A)"', "/" marking] & \mathcal{S}\!\mathit{et}
\end{tikzcd}$$

4 Mosaics and ribbon product

In this section is introduced the key notion of *weft mosaics*. They form a new tool for specification, which generalizes wefts (hence algebraic specifications).

In 2 we used a ribbon product in order to amplify a weft of ambigraphs $\mathbf{S}$ according to a counter-model κ of $\mathbf{E}_{\mathcal{A}mbi}$. This was not sufficient for our purpose, because all arrows of $\mathbf{S}$ were amplified in the same way. This may lead us to first *classify* the ingredients of $\mathbf{S}$, before amplifying each of them according to its class. Such a classification could be done by an indexation, as defined in [guide1], however this is not sufficient for most applications. Here we define *stratifications*, which generalize indexations, and we use them for classifying the ingredients of $\mathbf{S}$.

Hence a *mosaic* is made of a weft $\mathbf{S}$ (called its *apparent weft*) together with a stratification (which classifies the ingredients of $\mathbf{S}$) and a counter-model κ (which gives the shape of the amplifications). In this way, a mosaic is able to specify implicit features while preserving their implicit nature. The *realizations* of a mosaic $\mathbf{\Sigma}$ take into account all its components. They are quite different from the realizations of its apparent weft $\mathbf{S}$. However it is possible, thanks to a *ribbon product*, to build a weft $\mathbf{S}^{ex}$ (called the *explicit weft* of $\mathbf{\Sigma}$) with the same realizations as $\mathbf{\Sigma}$. It means that it is possible to *explicit* a mosaic.

First in 4.1 we define the *stratifications*. Then in 4.2, using the crown product and the stratifications, we define the ribbon product and we state its fundamental property. The mosaics and their realizations are defined in 4.3, and the fact that it is possible to explicit any mosaic is obtained as an immediate consequence of the fundamental property of the ribbon product. These definitions and results will be illustrated in 5.

4.1 Stratifications

First, we recall the notions of indexation, fibration and blow-up, as defined in [guide1].

Let $\mathbf{E}$ be a projective sketch. Then each model ι of $\mathbf{E}$ determines a projective sketch $\mathbf{E}\backslash\iota$ called the *blow-up* and a homomorphism of projective sketches $\mathbf{E}\backslash\!\!\backslash\iota$ called the *fibration*:

$$\mathbf{E}\backslash\!\!\backslash\iota : \mathbf{E}\backslash\iota \to \mathbf{E} \; .$$

The definition of the fibration, in [guide1, section 4.2], is such that:

- for all point E of $\mathbf{E}$, the inverse image of E by $\mathbf{E}\backslash\!\!\backslash\iota$ is made of the points $[E,i]$, where i is in the set $\iota(E)$;
- for all arrow $e : E_1 \to E_2$ of $\mathbf{E}$, the inverse image of e by $\mathbf{E}\backslash\!\!\backslash\iota$ is made of the arrows $[e,i_1] : [E_1,i_1] \to [E_2,i_2]$, where i_1 is in the set $\iota(E_1)$ and $i_2 = \iota(e)(i_1) \in \iota(E_2)$;
- and so on...

Each *ι-indexation*, *i.e.* each $h : \mu \to \iota$ in $\mathcal{M}od(\mathbf{E})$, determines, by the fundamental theorem on the blow-up (see [guide1, section 4.2]), a model $\iota/\!\!\backslash\!\!\backslash h$ of $\mathbf{E}\backslash\iota$:

$$\iota/\!\!\backslash\!\!\backslash h : \mathbf{E}\backslash\iota \to \mathcal{S}et \, ,$$

such that, for all point E of $\mathbf{E}$ and all $i \in \iota(E)$:

$$(\iota/\!\!\backslash\!\!\backslash h)([E,i]) = \{x \in \mu(E) \mid h(E)(x) = i\} \subseteq \mu(E) \, ,$$

and for all arrow $e : E_1 \to E_2$ of $\mathbf{E}$ and all $i \in \iota(E_1)$:

$$(\iota/\!\backslash\!\backslash h)([e,i]) : (\iota/\!\backslash\!\backslash h)[E_1, i_1] \to (\iota/\!\backslash\!\backslash h)[E_2, \iota(e)(i_1)] \text{ is the restriction of } \mu(e) : \mu(E_1) \to \mu(E_2)\ .$$

Moreover, the inclusions $(\iota/\!\backslash\!\backslash h)([E,i]) \subseteq \mu(E)$ define a homomorphism $\iota/\!\backslash\!\backslash\!\backslash h : \iota/\!\backslash\!\backslash h \to \mu \circ (\mathbf{E}\backslash\!\backslash\iota)$:

$$\begin{tikzcd}
\mathbf{E}\backslash\iota \arrow[rr, "\mathbf{E}\backslash\!\backslash\iota"] \arrow[dr, bend right, "\iota/\!\backslash\!\backslash h"'] \arrow[rr, phantom, shift right=3, "\overset{\iota/\!\backslash\!\backslash\!\backslash h}{\Longrightarrow}"] & & \mathbf{E} \arrow[dl, bend left, "\mu"] \\
& \mathcal{S}et &
\end{tikzcd}$$

More generally (see [guide1, section 4.3]), each *ι-indexation* $H : \mathbf{S} \to \iota$ of a $\mathcal{M}od(\mathbf{E})$-weft $\mathbf{S}$ determines a $\mathcal{M}od(\mathbf{E}\backslash\iota)$-weft $\iota/\!\backslash\!\backslash H$:

$$\iota/\!\backslash\!\backslash H : \mathbf{E}\backslash\iota \overset{\frown}{\longrightarrow} \mathcal{S}et\ ,$$

and a *loose* homomorphism $\iota/\!\backslash\!\backslash\!\backslash H : \iota/\!\backslash\!\backslash H \to \mathbf{S} \circ (\mathbf{E}\backslash\!\backslash\iota)$:

$$\begin{tikzcd}
\mathbf{E}\backslash\iota \arrow[rr, "\mathbf{E}\backslash\!\backslash\iota"] \arrow[dr, bend right, "(\ \iota/\!\backslash\!\backslash H"'] \arrow[rr, phantom, shift right=3, "\overset{\iota/\!\backslash\!\backslash\!\backslash H}{\Longrightarrow}"] & & \mathbf{E} \arrow[dl, bend left, ")\ \mathbf{S}"] \\
& \mathcal{S}et &
\end{tikzcd}$$

As noted in [guide1, section 4.3], whereas the homomorphisms of $\mathcal{M}od(\mathbf{E})$-wefts behave well with respect to the realizations, the loose homomorphisms do *not* behave well. And actually, this is the kind of property we are looking for: see the example in 2 and the discussion in 5.1.

It happens that the indexations are not general enough for most applications. This will be illustrated by example 2, and by the problem of natural numbers with a predecessor operation in 5. This is why we now define the stratifications.

Let us consider a homomorphism of projective sketches:

$$\rho : \mathbf{F} \to \mathbf{E}\ .$$

Definition 5 A *stratification along ρ*, or *ρ-stratification*, is a triple (μ, ν, r) (or just r) where:

- μ is a model of $\mathbf{E}$,
- ν is a model of $\mathbf{F}$,
- and $r : \nu \to \mu \circ \rho$ is a homomorphism of models of $\mathbf{F}$.

$$\begin{tikzcd}
\mathbf{F} \arrow[rr, "\rho"] \arrow[dr, bend right, "\nu"'] \arrow[rr, phantom, shift right=3, "\overset{r}{\Longrightarrow}"] & & \mathbf{E} \arrow[dl, bend left, "\mu"] \\
& \mathcal{S}et &
\end{tikzcd}$$

A *homomorphism of ρ-stratifications*, from (μ_1, ν_1, r_1) towards (μ_2, ν_2, r_2), is a pair (h, k) where:

- $h : \mu_1 \to \mu_2$ is a homomorphism of models of $\mathbf{E}$,
- and $k : \nu_1 \to \nu_2$ is a homomorphism of models of $\mathbf{F}$,

such that:

$$r_2 \circ k = (h \circ \rho) \circ r_1\ .$$

In this way we get the category of ρ-stratifications. It can be proven that this category is projectively sketchable, by a *lax-colimit* of ρ: see the reference manual.

It is easy to see that a weft of ρ-stratifications determines on one hand (looking only at the "μ's" in the support and the constraints) a weft $\mathbf{S}$ of models of $\mathbf{E}$, and on the other hand (looking only at the "ν's" in the support and the constraints) a weft $\mathbf{T}$ of models of $\mathbf{F}$. Moreover, $\mathbf{T}$ is related to $\mathbf{S} \circ \rho$ by a family of homomorphisms of models of $\mathbf{F}$ (the "r's" in the support and the constraints). These homomorphisms define a loose homomorphism of $\mathcal{M}od(\mathbf{F})$-wefts.

It follows that a weft of ρ-stratifications can be identified to a triple $(\mathbf{S}, \mathbf{T}, R)$ (or just R) where:

- $\mathbf{S}$ is a weft of models of $\mathbf{E}$,
- $\mathbf{T}$ is a weft of models of $\mathbf{F}$,
- and $R : \mathbf{T} \to \mathbf{S} \circ \rho$ is a loose homomorphism of $\mathcal{M}od(\mathbf{F})$-wefts.

$$\begin{tikzcd} \mathbf{F} \arrow[rr, "\rho"] \arrow[dr, bend right, "\mathbf{T}"'] & \arrow[d, phantom, "\overset{R}{\Longrightarrow}"] & \mathbf{E} \arrow[dl, bend left, "\mathbf{S}"] \\ & \mathcal{S}et & \end{tikzcd}$$

The special case of the stratifications associated with an indexation suggests a method for the description of all the stratifications, using *inverse images*, which will be used in the examples.

To begin with, any homomorphism of projective sketches $\rho : \mathbf{F} \to \mathbf{E}$ can be described by inverse images. For this purpose, all the ingredients of the projective sketch $\mathbf{E}$ are first described. Then:

- for all point E of $\mathbf{E}$ we describe all the points F of $\mathbf{F}$ *above* E, *i.e.* such that their image by ρ is E; they can be denoted $[E, i]$ where i is in some suitable set;
- for all arrow e of $\mathbf{E}$ we describe all the arrows f of $\mathbf{F}$ *above* e, *i.e.* such that their image by ρ is e; they can be denoted $[e, j]$ where j is in some suitable set;
- for all distinguished projective cone c of $\mathbf{E}$ we describe all the distinguished projective cones d of $\mathbf{F}$ *above* c, *i.e.* such that their image by ρ is c; they can be denoted $[c, k]$ where k is in some suitable set;
- and so on...

In this way, the homomorphism ρ is known: indeed, $\rho([E, i]) = E$, $\rho([e, j]) = e$, *etc.*

Of course, whenever ρ is not a fibration, there is no simple relation between the set of the j's in the description of the arrows $[e, j]$ above an arrow $e : E_1 \to E_2$ and the set of the i_1's in the description of the points $[E_1, i_1]$ above the point E_1.

Now, a ρ-stratification $(\mu, \nu, r : \nu \to \mu \circ \rho)$ can also be described by inverse images. First the model μ of $\mathbf{E}$ is described, which means that a set $\mu(E)$ is given for all point E of $\mathbf{E}$, and a map $\mu(e)$ for all arrow e of $\mathbf{E}$. Then:

- for all point E of $\mathbf{E}$ and all point $[E, i]$ of $\mathbf{F}$ above E, we describe the set $\nu([E, i])$;
- for all arrow e of $\mathbf{E}$ and all arrow $[e, j]$ of $\mathbf{F}$ above e, we describe the map $\nu([e, j])$;
- and for all point E of $\mathbf{E}$ and all point $[E, i]$ of $\mathbf{F}$ above E, we describe the map $r([E, i]) : \nu([E, i]) \to \mu(E)$.

Quite often in practice, the maps $r([E,i])$ are inclusions and each application $\nu([e,j])$ is the suitable restriction of $\mu(e)$. Then this description becomes much simpler, since the maps $\nu([e,j])$ and $r([E,i])$ are known. In this case it is sufficient to describe:

- for all point E of $\mathbf{E}$ and all point $[E,i]$ of $\mathbf{F}$ above E, the subset $\nu([E,i])$ of $\mu(E)$.

Whenever an ingredient x of μ of nature E is in $\nu([E,i])$, we say that i is *an index* for x, or that x *satisfies* i.

Moreover, under these assumptions, if there are several arrows $[e,j_k] : [E_1,i_1] \to [E_2,i_2]$ above the same arrow $e : E_1 \to E_2$ and with the same values of i_1 and i_2, they are all interpreted by ν as the same map. When there is only one such arrow $[e,j] : [E_1,i_1] \to [E_2,i_2]$, it is sometimes denoted:

$$[e, i_1 \Rightarrow i_2] : [E_1,i_1] \to [E_2,i_2] \ .$$

This notation refers to the property corresponding to the arrow $[e, i_1 \Rightarrow i_2]$: *if an ingredient x_1 of μ of nature E_1 satisfies i_1, then the ingredient $x_2 = \mu(e)(x_1)$ of μ of nature E_2 satisfies i_2.*

Example 1 The example of partial functions (where "partial" means "strictly partial", *i.e.* not total), without any constraint, has been considered in [guide1, section 4.1]. We have described the directed graph $\mathcal{I}^0_{part}$ and the homomorphism of projective sketches:

$$\rho_{part} : \mathbf{F}_{part} \to \mathbf{E}_{\mathcal{D}ir} \ ,$$

where $\mathbf{F}_{part}$ is the blow-up $\mathbf{E}_{\mathcal{D}ir} \backslash \mathcal{I}^0_{part}$ and ρ_{part} is the fibration $\mathbf{E}_{\mathcal{D}ir} \backslash\!\!\backslash \mathcal{I}^0_{part}$.

Let $\mathcal{F}unc$ denote the category with the sets for points and the functions (which may be either total or partial) for arrows. The $\mathcal{I}^0_{part}$-indexation h^0_{part} of $\mathcal{F}unc$ [guide1, section 4.1] determines a ρ_{part}-stratification $(\mathcal{F}unc, \mathcal{H}^0_{part}, r^0_{part})$, where:

$$\mathcal{H}^0_{part} = \mathcal{I}^0_{part} /\!\!\backslash\!\!\backslash h^0_{part} \text{ and } r^0_{part} = \mathcal{I}^0_{part} /\!\!\backslash\!\!\backslash\!\!\backslash h^0_{part} \quad :$$

$\mathbf{F}_{part}$-Model $\mathcal{H}^0_{part} = \mathcal{I}^0_{part} /\!\!\backslash\!\!\backslash h^0_{part}$:
image of $[\mathtt{Pt}, I]$: all the sets,
image of $[\mathtt{Ar}, tot]$: all the total fonctions,
image of $[\mathtt{Ar}, part]$: all the partial functions.

Stratification $r^0_{part} = \mathcal{I}^0_{part} /\!\!\backslash\!\!\backslash\!\!\backslash h^0_{part} : \mathcal{H}^0_{part} \to \mathcal{F}unc \circ \rho_{part}$:
image of $[\mathtt{Pt}, I]$: it is the identity,
image of $[\mathtt{Ar}, tot]$: it is the inclusion of the set of total functions in the set of functions,
image of $[\mathtt{Ar}, part]$: it is the inclusion of the set of partial functions in the set of functions.

$$\begin{tikzcd} \mathbf{F}_{part} \arrow[rr, "\rho_{part}"] \arrow[dr, bend right, "\mathcal{H}^0_{part}"'] & \arrow[d, phantom, "\Rightarrow r^0_{part}"] & \mathbf{E}_{\mathcal{D}ir} \arrow[dl, bend left, "\mathcal{F}unc"] \\ & \mathcal{S}et & \end{tikzcd}$$

In a similar way, the $\mathcal{I}^0_{part}$-indexation $h^0_{s,p}$ of $\mathcal{G}^0_{s,p}$ [guide1, section 4.1] determines a ρ_{part}-stratification $(\mathcal{G}^0_{s,p}, \mathcal{H}^0_{s,p}, r^0_{s,p})$, where:

$$\mathcal{H}^0_{s,p} = \mathcal{I}^0_{part} /\!\!\backslash\!\!\backslash h^0_{s,p} \text{ and } r^0_{s,p} = \mathcal{I}^0_{part} /\!\!\backslash\!\!\backslash\!\!\backslash h^0_{s,p} \quad :$$

$\mathbf{F}_{part}$-Model $\mathcal{H}^0_{s,p} = \mathcal{I}^0_{part} /\!\!\backslash h^0_{s,p}$:
image of $[\mathtt{Pt}, I]$: U, N,
image of $[\mathtt{Ar}, tot]$: $z : U \to N$, $s : N \to N$,
image of $[\mathtt{Ar}, part]$: $p : N \to N$.

Stratification $r^0_{s,p} = \mathcal{I}^0_{part} /\!\!\backslash\!\!\backslash h^0_{s,p} : \mathcal{H}^0_{s,p} \to \mathcal{G}^0_{s,p} \circ \rho_{part}$:
image of $[\mathtt{Pt}, I]$: $\{U, N\} \xrightarrow{\subseteq} \{U, N\}$,
image of $[\mathtt{Ar}, tot]$: $\{z, s\} \xrightarrow{\subseteq} \{z, s, p\}$,
image of $[\mathtt{Ar}, part]$: $\{p\} \xrightarrow{\subseteq} \{z, s, p\}$.

$$
\begin{tikzcd}
\mathbf{F}_{part} \arrow[rr, "\rho_{part}"] \arrow[dr, bend right, "\mathcal{H}^0_{s,p}"'] \arrow[rr, phantom, shift right=3, "\overset{r^0_{s,p}}{\Longrightarrow}"] & & \mathbf{E}_{\mathcal{D}ir} \arrow[dl, bend left, "\mathcal{G}^0_{s,p}"] \\
& \mathcal{S}et &
\end{tikzcd}
$$

The homomorphisms of ρ_{part}-stratifications from $r^0_{s,p}$ towards r^0_{part} express the fact that the interpretations of the arrows z and s of $\mathcal{G}^0_{s,p}$ should be total, while the interpretation of the arrow p should be partial.

Example 2 Example 1 shows that partial functions can be handled thanks to an indexation, at least as long as we do not try to compose the functions. In order to take care of the composition of functions, we should extend $\mathcal{I}^0_{part}$ by adding to it all the composed arrows, including the arrow $part \circ tot$. When we compose a total function and a partial function we generally get a partial function. Henceforth, it is natural to define $part \circ tot = part$. However, as has been noted in [guide1, section 4.1], this does not fit, because it may happen that the composition of a total function and a partial function is a total function: see $pred \circ succ : \mathbb{N} \to \mathbb{N}$.

In order to solve this problem, first we weaken our requirements, and then we use a stratification.

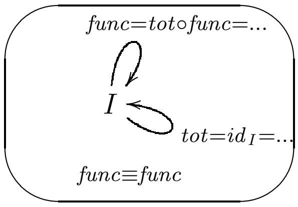
Our requirements are weakened in the following way: for all arrow g of a directed graph $\mathcal{G}$, it can still be said that the interpretation of g should be a total function, but it cannot be said that it should be a partial function. It means that the arrows of $\mathcal{G}$ can be indexed by two symbols: *tot* for an interpretation as a total function, and *func* otherwise. Whereas the properties "total" and "partial" (corresponding to the indices *tot* and *part* in example 1) are not compatible, the properties "total" and "any" (corresponding here to the indices *tot* and *func*) are compatible: indeed each total function is a function. This last point cannot be expressed by an indexation, but we will see that it can be expressed by a stratification.

However, as in example 1, let us first consider an ambigraph $\mathcal{I}_{func}$, the blow-up $\mathbf{E}_{\mathcal{A}mbi} \backslash \mathcal{I}_{func}$ and the fibration $\mathbf{E}_{\mathcal{A}mbi} \backslash\!\!\backslash \mathcal{I}_{func}$. The ambigraph $\mathcal{I}_{func}$ is the same, apart from the name of the arrow *func*, as the ambigraph $\mathcal{I}_{part}$ in [guide1, section 4.1].

Ambigraph $\mathcal{I}_{func}$:

point:	I,
arrows:	$func : I \to I,\ tot : I \to I$,
identity arrow:	$tot = id_I$,
composed arrows:	$tot \circ tot = tot,\ func \circ func = func \circ tot = tot \circ func = func$,
equations:	$func \equiv func,\ tot \equiv tot$.

$\mathcal{I}^0_{part}$:

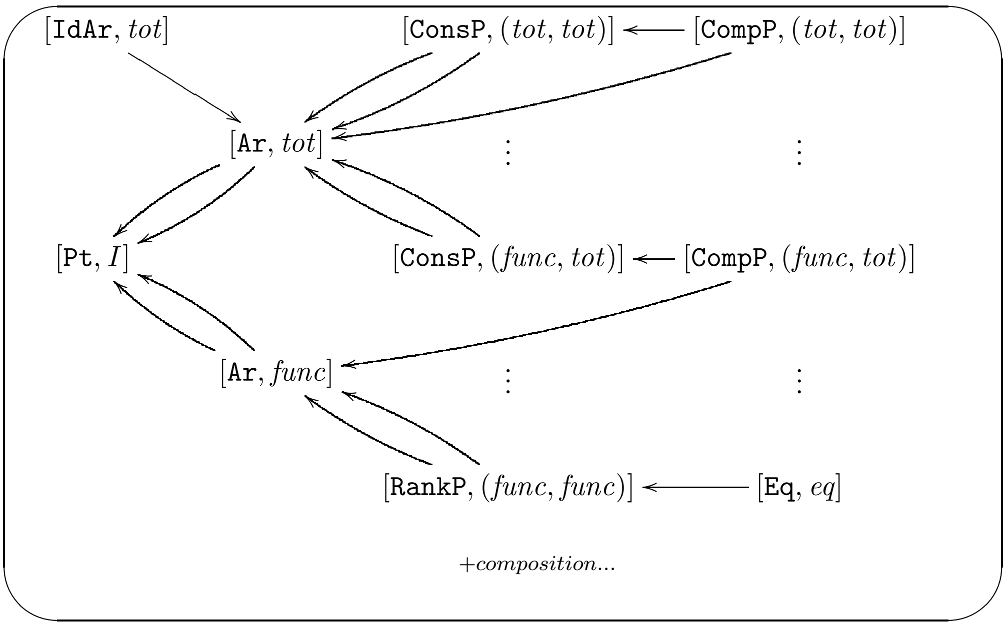


Projective sketch $\mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I}_{func}$:

(and homomorphism $\mathbf{E}_{\mathcal{A}mbi}\backslash\!\backslash\mathcal{I}_{func} : \mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I}_{func} \to \mathbf{E}_{\mathcal{A}mbi}$):

points: $[\mathtt{Pt}, I]$, $[\mathtt{Ar}, func]$, $[\mathtt{Ar}, tot]$, $[\mathtt{IdAr}, tot]$,
$[\mathtt{ConsP}, (i_1, i_2)]$ and $[\mathtt{CompP}, (i_1, i_2)]$ for all i_1 and i_2 in $\{func, tot\}$,
$[\mathtt{RankP}, (i_g, i_d)]$ for all i_g and i_d in $\{func, tot\}$,
$[\mathtt{Eq}, func \equiv func]$,

arrows: $[\mathtt{dom}, func]$, $[\mathtt{codom}, func] : [\mathtt{Ar}, func] \to [\mathtt{Pt}, I]$,
$[\mathtt{dom}, tot]$, $[\mathtt{codom}, tot] : [\mathtt{Ar}, tot] \to [\mathtt{Pt}, I]$,
etc.



Our aim is to generalize example 1 in the following way. The point $[\mathtt{Pt}, I]$ will be interpreted as the "set of sets", and the points $[\mathtt{Ar}, tot]$ and $[\mathtt{Ar}, func]$ respectively as the set of total functions and the set of all functions. In order to take into account the fact that each total function is a function, $\mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I}_{func}$ is extended by an arrow from $[\mathtt{Ar}, tot]$ towards $[\mathtt{Ar}, func]$, which will be interpreted as the inclusion of the set of total functions into the set of all functions. In other words, we are building a homomorphism of projective sketches:

$$\rho_{func} : \mathbf{F}_{func} \to \mathbf{E}_{func} \ ,$$

which extends the fibration $\mathbf{E}_{\mathcal{A}mbi}\backslash\!\backslash\mathcal{I}_{func}$, but which is no more a fibration. There are only minor differences between $\mathbf{E}_{\mathcal{A}mbi}$ and $\mathbf{E}_{func}$, in particular their models are the same ones. On the contrary, $\mathbf{F}_{func}$ is significantly different from $\mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I}_{func}$.

PPROJECTIVE SKETCH $\mathbf{E}_{func}$:
extends: $\mathbf{E}_{\mathcal{A}mbi}$,
identity arrow: $id_{\mathtt{Ar}} : \mathtt{Ar} \to \mathtt{Ar}$.

PROJECTIVE SKETCH $\mathbf{F}_{func}$
(and HOMOMORPHISM $\rho_{func} : \mathbf{F}_{func} \to \mathbf{E}_{func}$):
extends: $\mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I}_{func}$ (and $\mathbf{E}_{\mathcal{A}mbi}\backslash\!\backslash\mathcal{I}_{func}$),
arrow: $[id_{\mathtt{Ar}}, tot \Rightarrow func] : [\mathtt{Ar}, tot] \to [\mathtt{Ar}, func]$.

Although it is above the identity arrow $id_{\mathtt{Ar}}$, the arrow $[id_{\mathtt{Ar}}, tot \Rightarrow func]$ is *not* an identity arrow.

The notation "$tot \Rightarrow func$" refers to the property: *if a function satisfies tot, then it will also satisfie func.*

Let us consider the ρ_{func}-stratification $(\mathcal{F}unc, \mathcal{H}_{func}, r_{func})$:

$\mathbf{F}_{func}$-MODEL $\mathcal{H}_{func}$:

image of $[\mathtt{Pt}, I]$:	all the sets,
image of $[\mathtt{Ar}, tot]$:	all the total functions,
image of $[\mathtt{Ar}, func]$:	all the functions,
image of $[id_{\mathtt{Ar}}, tot \Rightarrow func]$:	the inclusion of the set of total functions in the set of functions,
image of $[\mathtt{Eq}, func \equiv func]$:	the equalities $f = f$ for all function f,
etc.	

STRATIFICATION $r_{func} : \mathcal{H}_{func} \to \mathcal{F}unc \circ \rho_{func}$:

image of $[\mathtt{Pt}, I]$:	the identity of the set of sets,
image of $[\mathtt{Ar}, tot]$:	the inclusion of the set of total functions in the set of functions,
image of $[\mathtt{Ar}, func]$:	the identity of the set of functions,
etc.	

$$\begin{tikzcd} \mathbf{F}_{func} \arrow[rr, "\rho_{func}"] \arrow[dr, bend right, "\mathcal{H}_{func}"'] & \arrow[d, phantom, "\overset{r_{func}}{\Longrightarrow}"] & \mathbf{E}_{func} \arrow[dl, bend left, "\mathcal{F}unc"] \\ & \mathcal{S}et & \end{tikzcd}$$

Now let us consider the ambigraph $\mathcal{G}_{s,p}$:

AMBIGRAPH $\mathcal{G}_{s,p}$:
extends: $\mathcal{G}^0_{s,p}$,
equation: $p \circ s \equiv id_N$.

In order to say that each arrow of $\mathcal{G}_{s,p}$ should be interpreted as a function, and that in addition the interpretations of the arrows z and s should be total functions, we consider the following ρ_{func}-stratification $(\mathcal{G}_{s,p}, \mathcal{H}_{s,p}, r_{s,p})$:

$\mathbf{F}_{func}$-MODEL $\mathcal{H}_{s,p}$:

image of $[\mathtt{Pt}, I]$: $\{U\ ,\ N\}$,
image of $[\mathtt{Ar}, tot]$: $\{z\ ,\ s\ ,\ id_N\}$,
image of $[\mathtt{Ar}, func]$: $\{z\ ,\ s\ ,\ id_N\ ,\ p\ ,\ p \circ s\}$,
image of $[id_{\mathtt{Ar}}, tot \Rightarrow func]$: the inclusion ,
image of $[\mathtt{Eq}, func \equiv func]$: $\{p \circ s \equiv id_N\}$,
etc.

STRATIFICATION $r_{s,p} : \mathcal{H}_{s,p} \to \mathcal{G}_{s,p} \circ \rho_{func}$:
image of $[\mathtt{Pt}, I]$: the identity of $\{U, N\}$,
image of $[\mathtt{Ar}, tot]$: the inclusion $\{z, s, id_N\} \xrightarrow{\subseteq} \{z, s, id_N, p, p \circ s\}$,
image of $[\mathtt{Ar}, func]$: the identity of $\{z, s, id_N, p, p \circ s\}$,
image of $[\mathtt{Eq}, func \equiv func]$: the identity of $\{p \circ s \equiv id_N\}$.
etc.

$$\begin{tikzcd}
\mathbf{F}_{func} \arrow[rr, "\rho_{func}"] \arrow[dr, bend right, "\mathcal{H}_{s,p}"'] & {} \arrow[d, phantom, "\overset{r_{s,p}}{\Longrightarrow}"] & \mathbf{E}_{func} \arrow[dl, bend left, "\mathcal{G}_{s,p}"] \\
& \mathcal{S}et &
\end{tikzcd}$$

4.2 Ribbon product

The definition of the ribbon product follows directly from the definitions of the crown product and the stratifications.

Thanks to the crown product, we may define the interpretations of a weft which assign to each point a set with some given property (the same property for all points), to each arrow a map with some given property (the same property for all arrows), and so on... If a stratification is used before the crown product, it becomes possible to assign sets with distinct properties to distinct points, maps with distinct properties to distinct arrows, and so on...: see 5 for an example.

We consider a homomorphism of projective sketches:

$$\rho : \mathbf{F} \to \mathbf{E}\,,$$

a cocomplete category:

$$\mathcal{W}\,,$$

and a counter-model of $\mathbf{F}$ towards $\mathcal{W}$:

$$\kappa : \mathbf{F} \longmapsto \mathcal{W}\ .$$

Definition 6 Let $(\mu, \nu, r : \nu \to \mu \circ \rho)$ be a ρ-stratification. The *ribbon product* of κ by r over ρ is the point of $\mathcal{W}$:

$$\kappa \circledR_\rho r = \kappa \copyright_{\mathbf{F}} \nu\ .$$

It follows immediately from proposition 1 that:

Proposition 2 *For all ρ-stratification (μ, ν, r) there is a canonical bijection:*

$$\boxed{\Theta(\kappa \circledR_\rho r) \cong \mathrm{Hom}_{\mathcal{M}od(\mathbf{F})}(\nu, \Theta \circ \kappa)\ .}$$

$$
\begin{tikzcd}
\kappa \circledR_\rho r \in \mathcal{W} & \mathbf{F} \arrow[l, "\kappa"'] \arrow[rr, "\rho"] \arrow[dr, bend right, "\nu"'] \arrow[rr, phantom, shift right=4ex, "\overset{r}{\Longrightarrow}"] & & \mathbf{E} \arrow[dl, bend left, "\mu"] \\
 & & \mathcal{S}et \arrow[from=1-1, "\Theta"'] &
\end{tikzcd}
$$

Definition 7 Let $(\mathbf{S}, \mathbf{T}, R : \mathbf{T} \to \mathbf{S} \circ \rho)$ be a weft of ρ-stratifications. The *ribbon product* of κ by R above ρ is the $\mathcal{W}$-weft:

$$\kappa \circledR_\rho R = \kappa \copyright_{\mathbf{F}} \mathbf{T} \ .$$

It follows immediately from theorem 1 that:

Theorem 2 (Fundamental property of the ribbon product) *For all weft of ρ-stratifications $(\mathbf{S}, \mathbf{T}, R)$ there is a canonical bijection:*

$$\boxed{\Theta_{\mathcal{W}eft}(\kappa \circledR_\rho R) \cong \mathrm{Real}_{\mathcal{M}od(\mathbf{F})}(\mathbf{T}, \Theta \circ \kappa) \ .}$$

$$
\begin{tikzcd}
\kappa \circledR_\rho R \in \mathcal{W}eft(\mathcal{W}) \arrow[r, hook] \arrow[drr, bend right, "\Theta_{\mathcal{W}eft}"'] & \mathcal{W} \arrow[dr, "\Theta"'] \arrow[d, phantom, "="] & \mathbf{F} \arrow[l, "\kappa"'] \arrow[rr, "\rho"] \arrow[d, bend right, "\mathbf{T}"'] \arrow[rr, phantom, shift right=4ex, "\overset{R}{\Longrightarrow}"] & & \mathbf{E} \arrow[dll, bend left, "\mathbf{S}"] \\
 & {} & \mathcal{S}et & &
\end{tikzcd}
$$

When $\mathcal{W} = \mathcal{W}eft(\mathcal{A})$ for some category $\mathcal{A}$, It follows immediately from corollary 1 that:

Corollary 2 *Assume that $\mathcal{W} = \mathcal{W}eft(\mathcal{A})$ for some cocomplete category $\mathcal{A}$. For all point A of $\mathcal{A}$ and all weft of ρ-stratifications $(\mathbf{S}, \mathbf{T}, R)$ there is a canonical bijection:*

$$\boxed{\mathrm{Real}_{\mathcal{A}}(\langle \kappa \circledR_\rho R \rangle, A) \cong \mathrm{Real}_{\mathcal{M}od(\mathbf{F})}(\mathbf{T}, \mathrm{Real}_{\mathcal{A}}(\kappa(-), A)) \ .}$$

$$
\begin{tikzcd}
\langle \kappa \circledR_\rho R \rangle \in \mathcal{W}eft(\mathcal{A}) = \mathcal{W} \arrow[dr, "\mathrm{Real}_{\mathcal{A}}(-,A)"'] & \mathbf{F} \arrow[l, "\kappa"'] \arrow[rr, "\rho"] \arrow[d, bend right, "\mathbf{T}"'] \arrow[rr, phantom, shift right=4ex, "\overset{R}{\Longrightarrow}"] & & \mathbf{E} \arrow[dll, bend left, "\mathbf{S}"] \\
 & \mathcal{S}et & &
\end{tikzcd}
$$

4.3 Mosaics

The wefts are not powerful enough to deal with the implicit features of computer languages. The *mosaics* are much better suited, as will appear in 5. Here we define the mosaics, we prove that they generalize the wefts, and we see how the ribbon product can be used in order to associate with each mosaic a weft with the same meaning.

In this section:

$$\mathcal{A} \text{ is any category.}$$

Let $\mathbf{S}$ be an $\mathcal{A}$-weft and A a point of $\mathcal{A}$. We have defined in [guide1, section 2.5] the set $\mathrm{Real}_{\mathcal{A}}(\mathbf{S}, A)$ of realizations of $\mathbf{S}$ towards A. The $\mathcal{A}$-weft $\mathbf{S}$ specifies each of its realizations. These notions are now generalized.

Definition 8 An $\mathcal{A}$-*mosaic* $\mathbf{\Sigma}$ is made up of:

- a homomorphism of projective sketches $\rho : \mathbf{F} \to \mathbf{E}$,

- a weft of ρ-stratifications $(\mathbf{S}, \mathbf{T}, R : \mathbf{T} \to \mathbf{S} \circ \rho)$,
- and a counter-model $\kappa : \mathbf{F} \longrightarrow\!\!\!\!\!\!\!+\;\; \mathcal{W}\!\mathit{eft}(\mathcal{A})$.

$$\begin{tikzcd}
\mathcal{W}\!\mathit{eft}(\mathcal{A}) = \mathcal{W} & \mathbf{F} \arrow[l, "\kappa"', "\shortmid" marking] \arrow[rr, "\rho"] \arrow[dr, bend right, "\mathbf{T}"'] & \arrow[d, phantom, "\overset{R}{\Longrightarrow}"] & \mathbf{E} \arrow[dl, bend left, "\mathbf{S}"] \\
 & & \mathcal{S}et &
\end{tikzcd}$$

Then $\mathbf{S}$ is called the *apparent weft* of the $\mathcal{A}$-mosaic $\mathbf{\Sigma}$, and $\mathbf{\Sigma}$ is also called *an $\mathcal{A}$-mosaic of* $\mathbf{S}$.

Definition 9 Let $\mathbf{\Sigma} = (\rho : \mathbf{F} \to \mathbf{E}, R : \mathbf{T} \to \mathbf{S} \circ \rho, \kappa : \mathbf{F} \longrightarrow\!\!\!\!\!\!\!+\;\; \mathcal{W}\!\mathit{eft}(\mathcal{A}))$ be an $\mathcal{A}$-mosaic, and let A be a point of $\mathcal{A}$. A *realization of* $\mathbf{\Sigma}$ *towards* A is a realization of $\mathbf{T}$ towards $\mathrm{Real}_{\mathcal{A}}(\kappa(-), A)$:

$$\boxed{\mathrm{Real}_{\mathcal{A}}(\mathbf{\Sigma}, A) = \mathrm{Real}_{\mathcal{M}od(\mathbf{F})}(\mathbf{T}, \mathrm{Real}_{\mathcal{A}}(\kappa(-), A))\ .}$$

If $\mathbf{\Sigma}$ is an $\mathcal{A}$-mosaic, A a point of $\mathcal{A}$ and ω a realization of $\mathbf{\Sigma}$ towards A, then we say that $\mathbf{\Sigma}$ *specifies* ω *(with respect to $\mathcal{A}$ and A)*. As in [guide1, section 2.5], this is only half of what is needed for mosaics to be a good specification tool. The second half, related to programs, will be seen in [guide3]: indeed $\mathcal{A}$-mosaics will allow us to deal with imperative programming.

Now let us check that mosaics do generalize wefts. Let $\mathbf{S}$ be an $\mathcal{A}$-weft, and assume that $\mathcal{A}$ is projectively sketchable, *i.e.* $\mathcal{A} \simeq \mathcal{M}od(\mathbf{E})$ for some projective sketch $\mathbf{E}$. Since $\mathcal{A}$ is a subcategory of $\mathcal{W}\!\mathit{eft}(\mathcal{A})$, the Yoneda counter-model of $\mathbf{E}$ [guide1, section 4.4] $\mathcal{Y}_{\mathbf{E}} : \mathbf{E} \longrightarrow\!\!\!\!\!\!\!+\;\; \mathcal{A}$ determines a counter-model, also denoted $\mathcal{Y}_{\mathbf{E}}$, from $\mathbf{E}$ towards $\mathcal{W}\!\mathit{eft}(\mathcal{A})$. The Yoneda lemma for projective sketches states that for all point A of $\mathcal{A}$, the set $\mathrm{Real}_{\mathcal{A}}(\mathcal{Y}_{\mathbf{E}}(-), A) = \mathrm{Hom}_{\mathcal{A}}(\mathcal{Y}_{\mathbf{E}}(-), A)$ can be identified with A. Let $\mathbf{\Sigma}$ denote the mosaic made of:
- $id_{\mathbf{E}} : \mathbf{E} \to \mathbf{E}$,
- $id_{\mathbf{S}} : \mathbf{S} \to \mathbf{S} \circ id_{\mathbf{E}}$,
- $\mathcal{Y}_{\mathbf{E}} : \mathbf{E} \longrightarrow\!\!\!\!\!\!\!+\;\; \mathcal{W}\!\mathit{eft}(\mathcal{M}od(\mathbf{E}))$.

It follows from the Yoneda lemma that the set of realizations of $\mathbf{\Sigma}$ can be identified with the set of realizations of $\mathbf{S}$.

From now on, we assume that:

$$\mathcal{A} \text{ is a cocomplete category,}$$

so that the category $\mathcal{W}\!\mathit{eft}(\mathcal{A})$ is also cocomplete.

Definition 10 Let $\mathbf{\Sigma} = (\rho, R, \kappa)$ be an $\mathcal{A}$-mosaic, The *explicit weft* of $\mathbf{\Sigma}$ is the $\mathcal{A}$-weft:

$$\boxed{\mathcal{E}\!\mathit{xpl}(\mathbf{\Sigma}) = \langle \kappa \circledR_{\rho} R \rangle\ .}$$

Corollary 2 states that, for all mosaic $\mathbf{\Sigma}$, the explicit weft $\mathcal{E}\!\mathit{xpl}(\mathbf{\Sigma})$ has the same realizations as $\mathbf{\Sigma}$. It can be reformulated as:

Corollary 3 (Fundamental property of mosaics) *Let $\mathcal{A}$ be a cocomplete category and A a point of $\mathcal{A}$. Let $\mathbf{\Sigma}$ be an $\mathcal{A}$-mosaic and $\mathcal{E}\!\mathit{xpl}(\mathbf{\Sigma})$ its explicit weft. Then there is a canonical bijection:*

$$\boxed{\mathrm{Real}_{\mathcal{A}}(\mathbf{\Sigma}, A) \cong \mathrm{Real}_{\mathcal{A}}(\mathcal{E}\!\mathit{xpl}(\mathbf{\Sigma}), A)\ .}$$

So, from the point of view of realizations, something as elaborated as a mosaic can be replaced by a weft, which may look much simpler. However, the direct definition of the realizations of a mosaic (definition 9), is often much easier to use. Indeed, the explicit weft $\mathcal{E}xpl(\mathbf{\Sigma})$ is quite large, hence difficult to handle. Moreover, it is a mixture of all the components of the mosaic, which cannot be retrieved from $\mathcal{E}xpl(\mathbf{\Sigma})$, however interesting they are on their own.

Consequently, on one hand it is important to know how to build the explicit weft of a mosaic, since this yields a consistent framework for dealing simultaneously with the functional point of view and the implicit features in the computer languages. On the other hand, in practice it is generally *unwise* to build the explicit weft, it is much more advisable to deal directly with the mosaic.

Example 3 In example 1 we defined a homomorphism $\rho_{func} : \mathbf{F}_{func} \to \mathbf{E}_{func}$ of projective sketches and a homomorphism $r_{s,p} : \mathcal{H}_{s,p} \to \mathcal{G}_{s,p} \circ \rho_{func}$ of models of $\mathbf{F}_{func}$, which is of course a loose homomorphism of $\mathcal{M}od(\mathbf{F}_{func})$-wefts (without constraints). We now describe some $\mathcal{A}mbi$-wefts and some homomorphisms between them.

$\mathcal{A}$-Weft $\kappa([\mathtt{Pt}, I])$:
point: H .

$$\kappa([\mathtt{Pt}, I]) : \quad H$$

$\mathcal{A}$-Weft $\kappa([\mathtt{Ar}, tot])$:
points: H_1 , H_2 ,
arrow: $k : H_1 \to H_2$.

$$\kappa([\mathtt{Ar}, tot]) : \quad \begin{array}{c} H_1 \\ \downarrow k \\ H_2 \end{array}$$

$\mathcal{A}$-Weft $\kappa([\mathtt{Ar}, part])$:
points: H_1 , H_2 , H_1' ,
arrows: $k : H_1' \to H_2$, $h : H_1' \to H_1$,
constraint: h is a monomorphism .

$$\kappa([\mathtt{Ar}, part]) : \quad \begin{tikzcd} H_1' \arrow[r, tail, "h"] \arrow[dr, "k"'] & H_1 \\ & H_2 \end{tikzcd}$$

Homomorphism $\kappa([\mathtt{dom}, tot]) : \kappa([\mathtt{Pt}, I]) \to \kappa([\mathtt{Ar}, tot])$:
points: $H \mapsto H_1$.

Homomorphism $\kappa([\mathtt{dom}, part]) : \kappa([\mathtt{Pt}, I]) \to \kappa([\mathtt{Ar}, part])$:
points: $H \mapsto H_1$.

Homomorphism $\kappa([\mathtt{codom}, tot]) : \kappa([\mathtt{Pt}, I]) \to \kappa([\mathtt{Ar}, tot])$:
points: $H \mapsto H_2$.

Homomorphism $\kappa([\mathtt{codom}, part]) : \kappa([\mathtt{Pt}, I]) \to \kappa([\mathtt{Ar}, part])$:
points: $H \mapsto H_2$.

The set-valued realizations of $\kappa([\mathtt{Pt}, I])$ are the sets, and the set-valued realizations of $\kappa([\mathtt{Ar}, tot])$ (*resp.* of $\kappa([\mathtt{Ar}, part])$) are the total functions (*resp.* all the functions).

It is possible to complete this description in order to get a counter-model κ_{func} from $\mathbf{F}_{func}$ towards $\mathcal{W}eft(\mathcal{A}mbi)$. In this way we get a mosaic:

$$\mathbf{\Sigma}_{func} \;=\; (\; \rho_{func} : \mathbf{F}_{func} \to \mathbf{E}_{func} \,,\; r_{s,p} : \mathcal{H}_{s,p} \to \mathcal{G}_{s,p} \circ \rho_{func} \,,\; \kappa_{func} : \mathbf{F}_{func} \longrightarrow\!\!\!\!\!\!+\;\; \mathcal{W}eft(\mathcal{A}mbi) \;) \;.$$

The set-valued realizations of $\mathbf{\Sigma}_{func}$ are such that the interpretations of the arrows z and s of $\mathcal{G}_{s,p}$ are total functions, while the interpretation of the arrow p is any function.

The explicit weft $\mathcal{E}xpl(\mathbf{\Sigma}_{func})$ contains:

$\mathcal{A}mbi$-WEFT $\mathcal{E}xpl(\boldsymbol{\Sigma}_{func})$:

points:	H_U , H_N , H'_p , ...
arrows:	$k_z : H_U \to H_N$, $k_s : H_N \to H_N$,
	$k_p : H'_p \to H_N$, $h_p : H'_p \to H_N$, ...
constraints:	h_p is a monomorphism , ...
and so on...	

$$\mathcal{E}xpl(\boldsymbol{\Sigma}_{func}) : \quad \begin{tikzcd} & H_U \arrow[d, "k_z"] \\ H'_p \arrow[r, tail, "h_p"] \arrow[r, bend right, "k_p"'] & H_N \arrow[loop below, "k_s"] \end{tikzcd} \quad \text{(partial description)}$$

5 Mosaics and ribbon product: an example

This example is based, as in 2, on the problem of *natural numbers with a predecessor operation*. In 2 we used a weft and a crown product, which was not subtle enough. Here, we use a mosaic and the corresponding ribbon product, leading to a much better solution. More precisely, instead of the weft of ambigraphs **S** of 2, here we use an $\mathcal{A}mbi$-mosaic $\mathbf{\Sigma}$, such that **S** is the apparent weft of $\mathbf{\Sigma}$.

This section runs like section 2: the support of **S** is studied in 5.7, then its constraints in 5.8 and 5.9. The required interpretation of each ingredient of **S** is expressed more precisely here than in 2: this is done in 5.1 and 5.2.

5.1 Analysis

The weft of ambigraphs **S** is essentially the same as in 2.1: it has the arrows z, s, p, the identity arrow id_N and the composed arrow $p \circ s$. We add the composed arrow $p \circ z$, only in order to be able to speak about the properties of the interpretation of $p \circ z$.

$\mathcal{A}mbi$-Weft **S**:

points:	$U\,,\, N\,,$
arrows:	$z : U \to N\,,\; s : N \to N\,,\; p : N \to N\,,$
	$id_N : N \to N\,,\; p \circ s : N \to N\,,\; p \circ z : U \to N\,,$
equation:	$p \circ s \equiv id_N\,,$
terminal point:	$U\,,$
initiality constraint:	the interpretation of $(U \xrightarrow{z} N \xrightarrow{s} N)$ is initial
	among the $(X_0 \xrightarrow{x_0} X \xrightarrow{f} X)$, where X_0 is terminal.

The analysis is the same as in 2.1, it proves that the set of set-valued realizations of **S**:

$$\mathrm{Real}_{\mathcal{A}mbi}(\mathbf{S}, \mathcal{S}et)$$

is irrelevant. Moreover, in 2 we could not find any solution among the sets of realizations of **S** in another ambigraph:

$$\mathrm{Real}_{\mathcal{A}mbi}(\mathbf{S}, \ldots)\ .$$

Indeed, the set $\mathrm{Real}_{\mathcal{A}mbi}(\mathbf{S}, \mathrm{Real}_{\mathcal{A}mbi}(\kappa(-), \mathcal{S}et))$, which appeared as a fairly good candidate when we looked only at the support of **S**, failed to deal in a correct way with the constraints, as seen in 2.8 and 2.9.

Here we first build a mosaic $\mathbf{\Sigma}$ with apparent weft **S**. Then, we proceed as in 2, with the mosaic $\mathbf{\Sigma}$ instead of the weft **S**. We build two new sets of realizations corresponding exactly to our requirements:

1. on one hand the set of set-valued realizations of $\mathbf{\Sigma}$:

   $$\mathrm{Real}_{\mathcal{A}mbi}(\mathbf{\Sigma}, \mathcal{S}et)\ ,$$

2. and on the other hand the set of set-valued realizations of the explicit weft of $\mathbf{\Sigma}$, built thanks to a *ribbon product*:

   $$\mathrm{Real}_{\mathcal{A}mbi}(\mathcal{E}xpl(\mathbf{\Sigma}), \mathcal{S}et)\ .$$

With the former point of view we can see why it is important for the stratification in $\mathbf{\Sigma}$ to be a loose homomorphism (see 4.1). Indeed, since loose homomorphisms behave badly with respect to the realizations, we may expect that some realizations of the mosaic $\mathbf{\Sigma}$ will be relevant, although all the realizations of its apparent weft $\mathbf{S}$ are irrelevant. In contrast to 2, here we succeed in formalizing in a correct way the natural numbers with a predecessor operation.

5.2 Comments

As in 2.2, rather than the set-valued realizations of $\mathbf{S}$, we look at interpretations of $\mathbf{S}$ which satisfy some comments. In 2.2, our comments said that the interpretation of each point of $\mathbf{S}$ should be a set with an error element, and that the interpretation of each arrow of $\mathbf{S}$ should propagate the error. Here our comments are more accurate: the interpretation of some arrows is not allowed to create an error, while on the contrary the interpretation of other arrows should always return an error.

Given an arrow $g : G_1 \to G_2$ in $\mathbf{S}$, if $X'_1 = X_1 \sqcup \{\varepsilon_{X_1}\}$ denotes the interpretation of G_1 and $X'_2 = X_2 \sqcup \{\varepsilon_{X_2}\}$ the interpretation of G_2, then three distinct comments may be used for g:

- $K(\mathtt{Ar}, \mathit{for})$: like $K(\mathtt{Ar})$ in 2: the interpretation of g is a map $f' : X'_1 \to X'_2$ which *propagates the error*, which means that $f'(\varepsilon_1) = \varepsilon_2$ (*for* for "*forward*").

- $K(\mathtt{Ar}, \mathit{ok})$: the interpretation of g is a map $f' : X'_1 \to X'_2$ which *does not create any error*, which means that $f'(\varepsilon_1) = \varepsilon_2$ and $f'(x) \in X_2$, for all $x \in X_1$.

- $K(\mathtt{Ar}, \mathit{err})$: the interpretation of g is a map $f' : X'_1 \to X'_2$ which *always returns the error*, which means that $f'(\varepsilon_1) = \varepsilon_2$ and $f'(x) = \varepsilon_2$, for all $x \in X_1$.

These comments are related: if the interpretation of g satisfies $K(\mathtt{Ar}, \mathit{ok})$ or $K(\mathtt{Ar}, \mathit{err})$, then it will also satisfy $K(\mathtt{Ar}, \mathit{for})$.

A map which does not create any error will be called an *ok-map*, and a map which always returns the error will be called an *error-map*. This is similar to the notations in [Goguen 78], and indeed, as in [Goguen 78], we "believe that significant improvements in the art of programming can be achieved by treating errors in a more systematic manner". However, we do not completely agree with this paper when it advocates to "include all exceptional state behavior, especially error messages, directly in the specifications". From our point of view, these exceptional behaviors are not included in the apparent weft, but only in the explicit weft, which is not meant to be built, as explained in 4.2.

In order to have homogenous notations, the unique comment for the interpretation of a point G of $\mathbf{S}$ is denoted:

- $K(\mathtt{Pt}, \mathit{Err})$: as $K(\mathtt{Pt})$ in 2: the interpretation of a point G satisfies $K(\mathtt{Pt}, \mathit{Err})$ if it is made of a set $X' = X \sqcup \{\varepsilon_X\}$.

The unique comment for the interpretation of an equation $g_l \equiv g_r$ of $\mathbf{S}$ is denoted:

- $K(\mathtt{Eq}, \mathit{eq})$: the interpretation of an equation $g_l \equiv g_r$ satisfies $K(\mathtt{Eq}, \mathit{eq})$ if it is made of two maps f'_l and f'_r with the same rank $X'_1 \to X'_2$ such that $f'_l = f'_r$.

With these notations:

- the interpretation of the points U and N should satisfy $K(\mathtt{Pt}, \mathit{Err})$,

- the interpretation of the arrows z and s should satisfy $K(\mathtt{Ar}, \mathit{ok})$,

- the interpretation of the arrow p should satisfy $K(\mathtt{Ar}, \mathit{for})$,
- the interpretation of the arrow $p \circ z$ should satisfy $K(\mathtt{Ar}, \mathit{err})$,
- the interpretation of the equation $p \circ s \equiv id_N$ should satisfy $K(\mathtt{Eq}, \mathit{eq})$.

From which it follows immediately that:

- the interpretation of the arrow id_N should satisfy $K(\mathtt{Ar}, \mathit{ok})$,
- the interpretation of all the arrows should satisfy $K(\mathtt{Ar}, \mathit{for})$.

Hence the interpretation of a unique arrow of **S** may satisfy several comments, in a *compatible* way. For instance, the fact that z satisfies $K(\mathtt{Ar}, \mathit{ok})$ and $K(\mathtt{Ar}, \mathit{for})$ does *not* mean that the interpretation of z includes two maps (one ok-map, and another map which propagates the error). It means that the interpretation of z includes a unique map, which is an ok-map, and which *consequently* propagates the error.

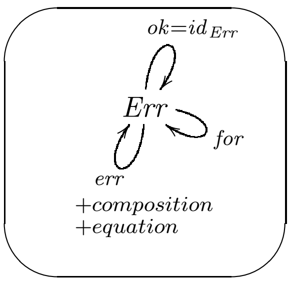
5.3 Homomorphism $\rho : \mathbf{F} \to \mathbf{E}$

The name of the properties used in the comments, *i.e.* *Err*, *for*, *ok*, *err* and *eq*, can be seen as the point, the arrows and the equation of an ambigraph $\mathcal{I}$:

AMBIGRAPH $\mathcal{I}$:

point:	Err ,
arrows:	$\mathit{for}\ ,\ \mathit{ok}\ ,\ \mathit{err} : \mathit{Err} \to \mathit{Err}$,
identity arrow:	$id_{\mathit{Err}} = \mathit{ok} : \mathit{Err} \to \mathit{Err}$,
composable pairs:	(i_1, i_2) for all i_1 and i_2 in $\{\mathit{ok}, \mathit{for}, \mathit{err}\}$,
composed arrows:	$\mathit{ok} \circ i = i \circ \mathit{ok} = i$ for all i (since $\mathit{ok} = id_{\mathit{Err}}$),
	$\mathit{err} \circ i = i \circ \mathit{err} = \mathit{err}$ for all i ,
	$\mathit{for} \circ \mathit{for} = \mathit{for}$,
equation:	$\mathit{for} \equiv \mathit{for}$, denoted eq .

$\mathcal{I}$:



The definition of the identity of *Err* corresponds to the fact that the identity map of a set with an error element is an ok-map.

The definition of the composition in $\mathcal{I}$ corresponds to the properties of composed maps. For instance, the equality $\mathit{err} \circ \mathit{for} = \mathit{err}$ means that, if (f'_1, f'_2) is a pair of consecutive maps such that f'_1 propagates the error and f'_2 is an error-map, then $f'_2 \circ f'_1$ is also an error-map.

Since $\mathcal{I}$, like any ambigraph, can be identified with a set-valued realization of the projective sketch $\mathbf{E}_{\mathcal{A}mbi}$, we can build the projective sketch blow-up of $\mathbf{E}_{\mathcal{A}mbi}$ by $\mathcal{I}$ and the corresponding fibration:

$$\mathbf{E}_{\mathcal{A}mbi}\backslash\!\!\backslash\mathcal{I} : \mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I} \to \mathbf{E}_{\mathcal{A}mbi} \ .$$

The projective sketch $\mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I}$ includes:

- above the point `Pt`: only the point $[\mathtt{Pt}, \mathit{Err}]$,
- above the point `Ar`: three points $[\mathtt{Ar}, \mathit{ok}]$, $[\mathtt{Ar}, \mathit{for}]$ and $[\mathtt{Ar}, \mathit{err}]$,
- above the arrow `dom`: three arrows $[\mathtt{dom}, i] : [\mathtt{Ar}, i] \to [\mathtt{Pt}, \mathit{Err}]$ for $i \in \{\mathit{ok}, \mathit{for}, \mathit{err}\}$,
- above the arrow `codom`: three arrows $[\mathtt{codom}, i] : [\mathtt{Ar}, i] \to [\mathtt{Pt}, \mathit{Err}]$ for $i \in \{\mathit{ok}, \mathit{for}, \mathit{err}\}$,
- above the point `IdAr`: the point $[\mathtt{IdAr}, \mathit{ok}]$,
- above the arrow $\mathtt{j}_{\mathtt{IdAr}}$: the arrow $[\mathtt{j}_{\mathtt{IdAr}}, \mathit{ok}] : [\mathtt{IdAr}, \mathit{ok}] \to [\mathtt{Ar}, \mathit{ok}]$,
- and so on...

$$\mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I} :\quad
\begin{tikzcd}
 & {[\mathtt{IdAr}, \mathit{ok}]} \arrow[dr, "{[\mathtt{j}_{\mathtt{IdAr}}, \mathit{ok}]}"] & & \cdots \\
 & & {[\mathtt{Ar}, \mathit{ok}]} \arrow[dll, bend right, "{[\mathtt{dom}, \mathit{ok}]}"'] \arrow[dll, "{[\mathtt{codom}, \mathit{ok}]}"] & \cdots \\
{[\mathtt{Pt}, \mathit{Err}]} & & {[\mathtt{Ar}, \mathit{for}]} \arrow[ll, bend right, "{[\mathtt{dom}, \mathit{for}]}"'] \arrow[ll, bend left, "{[\mathtt{codom}, \mathit{for}]}"] & \cdots \\
 & & {[\mathtt{Ar}, \mathit{err}]} \arrow[ull, "{[\mathtt{dom}, \mathit{err}]}"'] \arrow[ull, bend left, "{[\mathtt{codom}, \mathit{err}]}"] & \cdots
\end{tikzcd}$$

$$\mathbf{E}_{\mathcal{A}mbi}\backslash\!\backslash\mathcal{I} \downarrow$$

$$\mathbf{E}_{\mathcal{A}mbi} :\quad
\begin{tikzcd}
 & \mathtt{IdAr} \arrow[dr, "\mathtt{j}_{\mathtt{IdAr}}"] & & \cdots \\
\mathtt{Pt} & & \mathtt{Ar} \arrow[ll, bend right, "\mathtt{dom}"'] \arrow[ll, bend left, "\mathtt{codom}"] & \cdots
\end{tikzcd}$$

The blow-up $\mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I}$ deals with the fact that distinct arrows of $\mathbf{S}$ may have interpretations which satisfy distinct comments. But it does not deal with the fact that these comments are related. The relations between the comments state that each ok-map (*resp.* each error-map) does propagate the error. In order to deal with these relations, we extend $\mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I}$ and $\mathbf{E}_{\mathcal{A}mbi}\backslash\!\backslash\mathcal{I}$. We also extend $\mathbf{E}_{\mathcal{A}mbi}$, though in an innocuous way, without modifying its realizations (such an extension is called *conservative*).

It can be helpful to use the point of view of logic (already used in 4.1) for which each point of a projective sketch is a *property name* and each arrow is a *deduction rule.* Then the composition of arrows corresponds to *reasoning by deduction*, and each distinguished projective cone corresponds to a *conjonction* of properties. For example, in the projective sketch $\mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I}$, the point $[\mathtt{Ar}, \mathit{ok}]$ corresponds to the property:

- *a map is an ok-map*,

and the arrow $[\mathtt{comp}, (\mathit{ok}, \mathit{ok})] : [\mathtt{CompP}, (\mathit{ok}, \mathit{ok})] \to [\mathtt{Ar}, \mathit{ok}]$ corresponds to the deduction rule:

- *if both maps in a composable pair are ok-maps, then the composed map is also an ok-map.*

Let us translate the deduction rule:

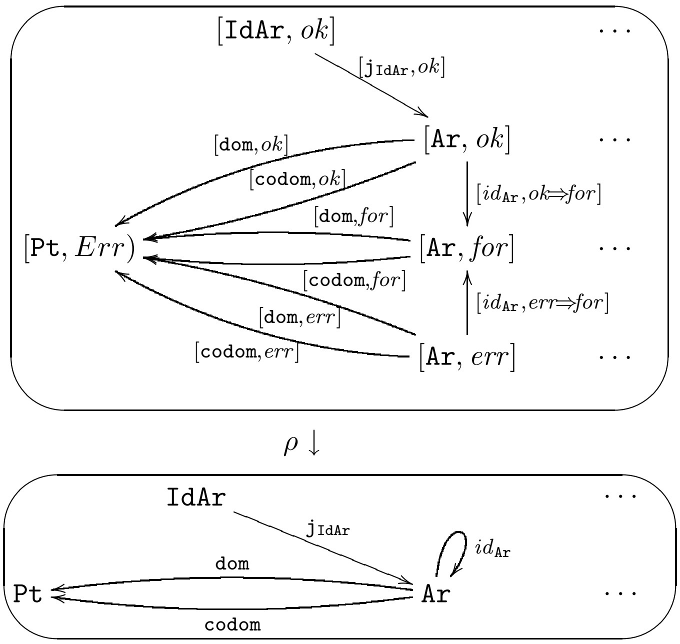
- *if the interpretation of an arrow is an ok-map (resp. an error-map), then it propagates the error.*

For this purpose, let $\mathbf{E}$ denote the projective sketch obtained by adding to $\mathbf{E}_{\mathcal{A}mbi}$ an identity arrow $id_{\mathtt{Ar}} : \mathtt{Ar} \to \mathtt{Ar}$ (clearly this is a conservative extension). Then, in order to translate the deduction rule, let us extend $\mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I}$ and $\mathbf{E}_{\mathcal{A}mbi}\backslash\!\!\backslash\mathcal{I}$ by adding:

- above $id_{\mathtt{Ar}}$: two arrows:
  $[id_{\mathtt{Ar}}, ok \Rightarrow for] : [\mathtt{Ar}, ok] \to [\mathtt{Ar}, for]$ and $[id_{\mathtt{Ar}}, err \Rightarrow for] : [\mathtt{Ar}, err] \to [\mathtt{Ar}, for]$.

Note that these arrows are *not* identities, and that this extension of $\mathbf{E}^{\mathcal{A}mbi}\backslash\!\!\backslash\mathcal{I}$ is *not* conservative. In this way we get a projective sketch $\mathbf{F}$ and a homomorphism $\rho : \mathbf{F} \to \mathbf{E}$ which extends the fibration $\mathbf{E}_{\mathcal{A}mbi}\backslash\!\!\backslash\mathcal{I} : \mathbf{E}_{\mathcal{A}mbi}\backslash\mathcal{I} \to \mathbf{E}_{\mathcal{A}mbi}$.

$\mathbf{F}$:



$\mathbf{E}$:

In 5.12 we will deal with other deduction rules.

5.4 Stratification (μ, ν, r)

Let us build a ρ-stratification, *i.e.* a set-valued realization μ of $\mathbf{E}$, a set-valued realization ν of $\mathbf{F}$ and a homomorphism $r : \nu \to \mu \circ \rho$ of set-valued realizations of $\mathbf{F}$.

As in 2.4 let:

$$\mu = \underline{\mathbf{S}} : \mathbf{E} \to \mathcal{S}et\ .$$

Then, for all point $[E, i]$ of $\mathbf{F}$, let us define the set $\nu([E, i])$ as the following subset of $\mu(E)$:

$$\nu([E, i]) = \text{ the set of the } x \in \mu(E) \text{ such that } x \text{ satisfies the comment } K(E, i)\ .$$

We get:

points: $\nu([\mathtt{Pt}, \mathit{Err}]) = \{U \,,\, N\}$,
arrows: $\nu([\mathtt{Ar}, \mathit{for}]) = \{z \,,\, s \,,\, p \,,\, id_N \,,\, p\circ s \,,\, p \circ z\}$,
$\nu([\mathtt{Ar}, \mathit{ok}]) = \{z \,,\, s \,,\, id_N\}$,
$\nu([\mathtt{Ar}, \mathit{for}]) = \{p \circ z\}$,
equation: $\nu([\mathtt{Eq}, \mathit{eq}]) = \{p\circ s \equiv id_N\}$,
etc.

For all arrow $[e,j] : [E_1, i_1] \to [E_2, i_2]$ of $\mathbf{F}$, let us define the map $\nu([e,j])$ as the restriction of $\mu(e)$:

$$\nu([e,j])(x) = \mu(e)(x) \text{ for all } x \in \nu([E_1, i_1]) \ .$$

It is easy to check that this is a map from $\nu([E_1, i_1])$ towards $\nu([E_2, i_2])$.

The inclusions $r([E,i]) : \nu([E,i]) \subseteq \mu(E)$ now define a homomorphism $r : \nu \to \mu\circ\rho$ of set-valued realizations of $\mathbf{F}$.

Thus, the triple (μ, ν, r) determines a ρ-stratification:

$$\begin{tikzcd}
\mathbf{F} \arrow[rr, "\rho"] \arrow[dr, bend right, "\nu"'] & \arrow[d, phantom, "\overset{r}{\Longrightarrow}"] & \mathbf{E} \arrow[dl, bend left, "\mu"] \\
& \mathcal{S}et &
\end{tikzcd}$$

5.5 Contravariant functor $\Theta : \mathcal{W} \mathrel{+\!\!\!\longrightarrow} \mathcal{S}et$

As in 2.5, let $\mathcal{W}$ denote the category of wefts of ambigraphs:

$$\mathcal{W} = \mathcal{W}\!\mathit{eft}(\mathcal{A}) \ \text{ where } \ \mathcal{A} = \mathcal{A}\mathit{mbi} \ ,$$

and let Θ denote the left-exact contravariant functor:

$$\Theta = \mathrm{Real}_{\mathcal{A}}(-, \mathcal{S}et) : \mathcal{W} \mathrel{+\!\!\!\longrightarrow} \mathcal{S}et \ .$$

5.6 Counter-model $\kappa : \mathbf{F} \mathrel{+\!\!\!\longrightarrow} \mathcal{W}$

As in 2.7, each comment $K(E,i)$ corresponds to a weft of ambigraphs $\kappa([E,i])$, in the sense that the interpretations which satisfy the comment $K(E,i)$ can be identified with the set-valued realizations of $\kappa([E,i])$.

$\mathcal{A}$-Weft $\kappa([\mathtt{Pt}, \mathit{Err}]) = \kappa(\mathtt{Pt})$.
$\mathcal{A}$-Weft $\kappa([\mathtt{Ar}, \mathit{for}]) = \kappa(\mathtt{Ar})$.
$\mathcal{A}$-Weft $\kappa([\mathtt{Ar}, \mathit{ok}])$:
extends: $\kappa([\mathtt{Ar}, \mathit{for}])$, ,
arrow: $k : H_1 \to H_2$, ,
equation: $k' \circ h_1 \equiv h_2 \circ k$ ..

$$\kappa([\mathtt{Ar}, \mathit{ok}]) : \quad \begin{tikzcd}
H_1 \arrow[r, "h_1"] \arrow[d, "k"] \arrow[dr, phantom, "\equiv"] & H_1' = H_1 + H_1^e \arrow[d, "k'"] \arrow[dr, phantom, "\equiv"] & H_1^e = \mathbb{1} \arrow[l, "h_1^e"'] \arrow[d, "k^e"] \\
H_2 \arrow[r, "h_2"] & H_2' = H_2 + H_2^e & H_2^e = \mathbb{1} \arrow[l, "h_2^e"']
\end{tikzcd}$$

$\mathcal{A}$-WEFT $\kappa([\mathtt{Ar}, err])$:
extends: $\kappa([\mathtt{Ar}, for])$,
arrow: $k'^e : H'_1 \to H^e_2$,
equation: $k' \equiv h^e_2 \circ k'^e$.

$$\kappa([\mathtt{Ar}, err]) : \begin{tikzcd} H_1 \arrow[r, "h_1"] & H'_1 = H_1 + H^e_1 \arrow[d, "k'"] \arrow[dr, "k'^e"] \arrow[dr, phantom, "\equiv"'] & H^e_1 = \mathbb{1} \arrow[l, "h^e_1"'] \arrow[d, "k^e"] \\ H_2 \arrow[r, "h_2"] & H'_2 = H_2 + H^e_2 & H^e_2 = \mathbb{1} \arrow[l, "h^e_2"'] \end{tikzcd}$$

Now it is easy to get a counter-model $\kappa : \mathbf{F} \longmapsto \mathcal{W}$ of the projective sketch $\mathbf{F}$ towards the category $\mathcal{W}$:

$$\begin{tikzcd} & \kappa([\mathtt{IdAr}, ok]) & & \cdots \\ & & \kappa([\mathtt{Ar}, ok]) \arrow[ul] & \cdots \\ \kappa([\mathtt{Pt}, Err]) \arrow[urr, bend left] \arrow[urr] \arrow[rr, bend left] \arrow[rr, bend right] \arrow[drr] \arrow[drr, bend right] & & \kappa([\mathtt{Ar}, for]) \arrow[u] \arrow[d] & \cdots \\ & & \kappa([\mathtt{Ar}, err]) & \cdots \end{tikzcd}$$

For all $i \in \{for, ok, err\}$:
HOMOMORPHISM $\kappa([\mathtt{dom}, i]) : \kappa([\mathtt{Pt}, Err) \to \kappa([\mathtt{Ar}, i])$:
lines: $\kappa([\mathtt{Pt}, Err]) \mapsto \kappa([\mathtt{Pt}, Err])_1$.
HOMOMORPHISM $\kappa([\mathtt{codom}, i]) : \kappa([\mathtt{Pt}, Err) \to \kappa([\mathtt{Ar}, i])$:
lines: $\kappa([\mathtt{Pt}, Err]) \mapsto \kappa([\mathtt{Pt}, Err])_2$.
For all $i \in \{ok, err\}$:
HOMOMORPHISM $\kappa([id_{\mathtt{Ar}}, i \Rightarrow for]) : \kappa([\mathtt{Ar}, for]) \to \kappa([\mathtt{Ar}, i])$:
it is the extension.
$\mathcal{A}$-WEFT $\kappa([\mathtt{RankP}, (for, for)]) = \kappa(\mathtt{RankP})$.
$\mathcal{A}$-WEFT $\kappa([\mathtt{Eq}, eq]) = \kappa(\mathtt{Eq})$.

And so on... In this way we get a counter-model:

$$\kappa : \mathbf{F} \longmapsto \mathcal{W} .$$

5.7 Ribbon product $\kappa \circledR_\rho r$

From the study above:

- the interpretation of an ingredient x of ν of nature $[E, i]$ (for all point $[E, i]$ of $\mathbf{F}$) should be an element of the set $\mathrm{Real}_{\mathcal{A}}(\kappa([E, i]), \mathcal{S}et)$. In addition, if x is in $\nu([E, i])$ for several values of i, then these interpretations should be compatible.

Now this study can be carried on in two ways:

1. By composition of the counter-model $\kappa : \mathbf{F} \longrightarrow \mathcal{W}$ and the contravariant functor $\Theta : \mathcal{W} \longrightarrow \mathcal{S}et$ (which is left-exact), we get a set-valued realization of $\mathbf{F}$:

$$\Theta \circ \kappa = \mathrm{Real}_{\mathcal{A}}(\kappa(-), \mathcal{S}et) : \mathbf{F} \to \mathcal{S}et\,,$$

which interprets each point $[E,i]$ of $\mathbf{F}$ as the set $\mathrm{Real}_{\mathcal{A}mbi}(\kappa([E,i]), \mathcal{S}et)$ of set-valued realizations of the weft of ambigraphs $\kappa([E,i])$.

From the study above, the natural numbers with the predecessor map define a homomorphism from ν to $\Theta \circ \kappa$, *i.e.* an element of the set:

$$\boxed{\mathrm{Hom}_{\mathcal{M}od(\mathbf{F})}(\nu, \Theta \circ \kappa)\ .}$$

Precisely, as in 2:
points: $U \mapsto \mathbb{U}' = \mathbb{U} + \{\varepsilon_{\mathbb{U}}\}\,,\ N \mapsto \mathbb{N}' = \mathbb{N} + \{\varepsilon_{\mathbb{N}}\}\,,$
arrows: $z \mapsto 0'\,,\ s \mapsto succ'\,,\ p \mapsto pred'\ .$

2. With the second point of view we consider for all point U and N of μ a copy of $\kappa([\mathtt{Pt}, Err])$, for all arrow z and s a copy of $\kappa([\mathtt{Ar}, ok])$ and a copy of $\kappa([\mathtt{Ar}, for])$, for the arrow p a copy of $\kappa([\mathtt{Ar}, for])$, for the arrow $p \circ z$ a copy of $\kappa([\mathtt{Ar}, err])$ and a copy of $\kappa([\mathtt{Ar}, for])$, and so on. Then we merge these wefts $\kappa([E,i])$ together according to the homomorphisms $\kappa([e,j])$. The weft of ambigraphs which we get is $\kappa \copyright_{\mathbf{F}} \nu$. By definition, this is the ribbon product:

$$\kappa \circledR_\rho r\ .$$

$\mathcal{A}$-Weft $\kappa \circledR_\rho r$:
lines: $\kappa([\mathtt{Pt}, Err])_U\,,\ \kappa([\mathtt{Pt}, Err])_N\,,$
arrows: $k_z : H_U \to H_N\,,\ k'_z : H'_U \to H'_N\,,\ k^e_z : H^e_U \to H^e_N\,,$
$k_s : H_N \to H_N\,,\ k'_s : H'_N \to H'_N\,,\ k^e_s : H^e_N \to H^e_N\,,$
$k'_p : H'_N \to H'_N\,,\ k^e_p : H^e_N \to H^e_N\,,$
$k'^e_{p\circ z} : H'_U \to H^e_N\,,$
equations: $k'_z \circ h_U \equiv h_N \circ k_z\,,\ k'_z \circ h^e_U \equiv h^e_N \circ k^e_z\,,$
$k'_s \circ h_N \equiv h_N \circ k_s\,,\ k'_s \circ h^e_N \equiv h^e_N \circ k^e_s\,,$
$k'_p \circ h^e_N \equiv h^e_N \circ k^e_p\,,$
$k'_p \circ k'_z \equiv h^e_N \circ k'^e_{p\circ z}\,,$
$k'_p \circ k'_s \equiv id_{H'_N}\ .$

$$\begin{tikzcd}
H_U \arrow[r, "h_U"] \arrow[d, "k_z"'] & H'_U = H_U + H^e_U \arrow[d, "k'_z"'] \arrow[dr, "k'^e_{p\circ z}"] & H^e_U = \mathbb{1} \arrow[l, "h^e_U"'] \arrow[d, "k^e_z"'] \\
H_N \arrow[r, "h_N"] \arrow[loop below, "k_s"] & H'_N = H_N + H^e_N \arrow[loop below, "k'_s"] \arrow[loop below, "k'_p"'] & H^e_N = \mathbb{1} \arrow[l, "h^e_N"'] \arrow[loop below, "k^e_s"] \arrow[loop below, "k^e_p"']
\end{tikzcd}$$

+equations...

In this example, the homomorphisms $\kappa([id_{\mathtt{Ar}}, ok \Rightarrow for])$ and $\kappa([id_{\mathtt{Ar}}, err \Rightarrow for])$ are extensions. Consequently, when we consider (for instance) for the arrow z a copy of

$\kappa([\mathtt{Ar}, ok])$ and a copy of $\kappa([\mathtt{Ar}, for])$, and when we merge them together according to $\kappa([id_{\mathtt{Ar}}, ok \Rightarrow for])$, the result is just a copy of $\kappa([\mathtt{Ar}, ok])$. This is another way to say that if the interpretation of an arrow satisfies $K(\mathtt{Ar}, ok)$ then it also satisfies $K(\mathtt{Ar}, for)$.

From the study above, the natural numbers with the predecessor operation define a set-valued realization of $\kappa \circledR_\rho r$, *i.e.* an element of the set:

$$\boxed{\Theta(\kappa \circledR_\rho r)\ .}$$

Precisely:

$$\begin{array}{ll} \textit{points}: & H_U \mapsto \mathbb{U}\,,\ H'_U \mapsto \mathbb{U}'\,,\ H^e_U \mapsto \{\varepsilon_{\mathbb{U}}\}\,, \\ & H_N \mapsto \mathbb{N}\,,\ H'_N \mapsto \mathbb{N}'\,,\ H^e_N \mapsto \{\varepsilon_{\mathbb{N}}\}\,, \\ \textit{arrows}: & k_z \mapsto 0\,,\ k_s \mapsto \textit{succ}\,, \\ & k'_z \mapsto 0'\,,\ k'_s \mapsto \textit{succ}'\,,\ k'_p \mapsto \textit{pred}'\,, \\ & k^e_z \mapsto (\varepsilon_{\mathbb{U}} \mapsto \varepsilon_{\mathbb{N}})\,,\ k^e_s \mapsto id_{\{\varepsilon_{\mathbb{N}}\}}\,,\ k^e_p \mapsto id_{\{\varepsilon_{\mathbb{N}}\}}\,, \\ & k'^e_{poz} \mapsto (* \mapsto \varepsilon_{\mathbb{N}}, \varepsilon_{\mathbb{U}} \mapsto \varepsilon_{\mathbb{N}})\ . \end{array}$$

Proposition 2 proves that there is a canonical bijection between the two sets:

$$\boxed{\Theta(\kappa \circledR_\rho r) \cong \mathrm{Hom}_{\mathcal{M}od(\mathbf{F})}(\nu, \Theta \circ \kappa)\ ,}$$

which means that:

$$\mathrm{Real}_{\mathcal{A}mbi}(\kappa \circledR_\rho r, \mathcal{S}et) \cong \mathrm{Hom}_{\mathcal{M}od(\mathbf{F})}(\nu, \mathrm{Real}_{\mathcal{A}mbi}(\kappa(-), \mathcal{S}et))\ .$$

5.8 Terminal point constraint

Let us now consider the constraint "$U = \mathbb{1}$" of $\mathbf{S}$, which means that the interpretation of U should be a terminal point. Let:

$$\mathbf{S}_1 : \mathbf{E} \overset{\frown}{\longrightarrow} \mathcal{S}et$$

be the weft of ambigraphs of support μ and constraint Γ (as in 2.8).

Let us decide that the ingredients of $\mu_{\mathcal{C}}$, $\mu_{\mathcal{D}}$ and $\mu_{\mathcal{U}}$ should satisfy:

- the interpretation of the points C (in $\mu_{\mathcal{C}}$, $\mu_{\mathcal{D}}$ and $\mu_{\mathcal{U}}$) and D (in $\mu_{\mathcal{D}}$ and $\mu_{\mathcal{U}}$) should satisfy $K(\mathtt{Pt}, Err)$,
- the interpretation of the arrow u (in $\mu_{\mathcal{U}}$) should satisfy $K(\mathtt{Ar}, ok)$, hence also $K(\mathtt{Ar}, for)$.

Then, proceeding as in 5.4 to build the ρ-stratification (μ, ν, r), we may build three more ρ-stratifications: $(\mu_{\mathcal{C}}, \nu_{\mathcal{C}}, r_{\mathcal{C}})$, $(\mu_{\mathcal{D}}, \nu_{\mathcal{D}}, r_{\mathcal{D}})$ and $(\mu_{\mathcal{U}}, \nu_{\mathcal{U}}, r_{\mathcal{U}})$. Their most significant part is the following one:

- $\nu_{\mathcal{C}}([\mathtt{Pt}, Err]) = \{C\}$ and $\nu_{\mathcal{C}}([E, i]) = \emptyset$ for the others $[E, i]$;
- $\nu_{\mathcal{D}}([\mathtt{Pt}, Err]) = \{C, D\}$ and $\nu_{\mathcal{D}}([E, i]) = \emptyset$ for the others $[E, i]$;
- $\nu_{\mathcal{U}}([\mathtt{Pt}, Err]) = \{C, D\}$, $\nu_{\mathcal{U}}([\mathtt{Ar}, ok]) = \nu_{\mathcal{U}}([\mathtt{Ar}, for]) = \{u\}$ and $\nu_{\mathcal{U}}([E, i]) = \emptyset$ for the others $[E, i]$.

The homomorphisms χ, γ and δ determine homomorphisms of ρ-stratifications, hence we get a constraint over the ρ-stratification r, and a weft $(\mathbf{S}_1, \mathbf{T}_1, R_1)$ of ρ-stratifications:

$$\begin{tikzcd}
\mathbf{F} \arrow[rr, "\rho"] \arrow[dr, bend right, "\mathbf{T}"'] \arrow[rr, phantom, shift right=3, "\overset{R}{\Longrightarrow}"] & & \mathbf{E} \arrow[dl, bend left, "\mathbf{S}"] \\
& \mathcal{S}et &
\end{tikzcd}$$

The $\mathcal{M}od(\mathbf{F})$-weft $\mathbf{T}_1$ has only one constraint, denoted Γ_1. Its potential is made of the extensions $\nu_{\mathcal{C}} \to \nu_{\mathcal{D}} \to \nu_{\mathcal{U}}$ and its body $\nu_{\mathcal{C}} \to \nu$ is defined by $C \mapsto U$.

As above, there are two points of view.

1. The first point of view considers the realizations of $\mathbf{T}_1$ towards $\Theta \circ \kappa$, *i.e.* the elements of the set:

   $$\boxed{\mathrm{Real}_{\mathcal{M}od(\mathbf{F})}(\mathbf{T}_1, \Theta \circ \kappa)\ .}$$

   They are the ambifunctors from ν to $\Theta \circ \kappa$ which satisfy the constraint Γ_1. It means that the interpretation of U by such a realization, denoted $X' = X \sqcup \{\varepsilon_X\}$, satisfies the following property: for all set $Y' = Y \sqcup \{\varepsilon_Y\}$ there is a unique map $f' : Y' \to X'$, such that $f'(\varepsilon_Y) = \varepsilon_X$ and $f'(Y) \subseteq X$. Equivalently, the interpretation of U should be, among the sets with an error, *terminal with respect to the ok-maps.* It is easy to check that X' should be a two-element set $\{*, \varepsilon_X\}$, *i.e.* X should be a one-element set $\{*\}$. This is *exactly* what we wish for the interpretation of U.

2. Similarly to $\kappa \circledR_\rho r$ above, we may build the three wefts of ambigraphs $\kappa \circledR_\rho r_{\mathcal{C}}$, $\kappa \circledR_\rho r_{\mathcal{D}}$ and $\kappa \circledR_\rho r_{\mathcal{U}}$, with the extensions:

   $$\kappa \circledR r_{\mathcal{C}} : \left(\begin{array}{c} \\ \bullet \rightarrow + \leftarrow \mathbb{I} \end{array}\right) \longrightarrow \kappa \circledR r_{\mathcal{D}} : \left(\begin{array}{c} \bullet \rightarrow + \leftarrow \mathbb{I} \\ \bullet \rightarrow + \leftarrow \mathbb{I} \end{array}\right) \longrightarrow \kappa \circledR r_{\mathcal{U}} : \left(\begin{array}{ccccc} \bullet & \rightarrow & + & \leftarrow & \mathbb{I} \\ \downarrow & \equiv & \downarrow & \equiv & \downarrow \\ \bullet & \rightarrow & + & \leftarrow & \mathbb{I} \end{array}\right)$$

   and the body:

   $$\langle \kappa \circledR_\rho \chi \rangle : \mathit{Supp}(\kappa \circledR_\rho r_{\mathcal{C}}) \to \mathit{Supp}(\kappa \circledR_\rho r)$$

   defined by:

   $$(H_C \xrightarrow{h_C} H'_C \xleftarrow{h^e_C} H^e_C) \quad \mapsto \quad (H_U \xrightarrow{h_U} H'_U \xleftarrow{h^e_U} H^e_U)\ .$$

   In this way we get the constraint $\langle \kappa \circledR_\rho \Gamma_1 \rangle$ over the ambigraph support of $\kappa \circledR_\rho r$. By definition, the weft of ambigraphs $\langle \kappa \circledR_\rho R_1 \rangle$ is made of $\kappa \circledR_\rho r$ and of the constraint $\langle \kappa \circledR_\rho \Gamma_1 \rangle$. Let us consider the set:

   $$\boxed{\Theta(\, \langle \kappa \circledR_\rho R_1 \rangle \,)}$$

   of set-valued realizations of $\langle \kappa \circledR_\rho R_1 \rangle$. It is easy to check that, as above, the interpretation of U by each of these realizations should be a two-element set, as we wish.

These two points of view are equivalent by corollary 2:

$$\boxed{\Theta(\, \langle \kappa \circledR_\rho R_1 \rangle \,) \cong \mathrm{Real}_{\mathcal{M}od(\mathbf{F})}(\mathbf{T}_1, \Theta \circ \kappa)\ ,}$$

or, equivalently:

$$\mathrm{Real}_{\mathcal{A}mbi}(\langle \kappa \circledR_\rho R_1\rangle, \mathcal{S}et) \cong \mathrm{Real}_{\mathcal{M}od(\mathbf{F})}(\mathbf{T}_1, \mathrm{Real}_{\mathcal{A}mbi}(\kappa(-), \mathcal{S}et)) \ .$$

In addition this gives *exactly*, for this example, what was required.

Let $\mathbf{\Sigma}_1$ denote the mosaic (ρ, R_1, κ). Its set-valued realizations have been defined as:

$$\mathrm{Real}_{\mathcal{A}mbi}(\mathbf{\Sigma}_1, \mathcal{S}et) = \mathrm{Real}_{\mathcal{M}od(\mathbf{F})}(\mathbf{T}_1, \mathrm{Real}_{\mathcal{A}mbi}(\kappa(-), \mathcal{S}et)) \ .$$

Then $\langle \kappa \circledR_\rho R_1\rangle$ is the explicit weft $\mathcal{E}xpl(\mathbf{\Sigma}_1)$ of $\mathbf{\Sigma}_1$. The equivalence of both points of view means that $\mathbf{\Sigma}_1$ and $\mathcal{E}xpl(\mathbf{\Sigma}_1)$ have the same set-valued realizations:

$$\mathrm{Real}_{\mathcal{A}mbi}(\mathcal{E}xpl(\mathbf{\Sigma}_1), \mathcal{S}et) \cong \mathrm{Real}_{\mathcal{A}mbi}(\mathbf{\Sigma}_1, \mathcal{S}et) \ .$$

5.9 Initiality constraint

Now let us handle, in the same two ways, the initiality constraint of $\mathbf{S}$.

Let us decide that the interpretation of each arrow in each ambigraph in the constraint should satisfy the comment $K(\mathtt{Ar}, ok)$. In this way we extend $(\mathbf{S}_1, \mathbf{T}_1, R_1)$ to a weft of ρ-stratifications $(\mathbf{S}, \mathbf{T}, R)$.

It is easy to check that we get the following result: the interpretation of N should be $\mathbb{N}' = \mathbb{N} \sqcup \{\varepsilon_{\mathbb{N}}\}$, and the interpretation of z and s should be the maps $0'$ and $succ'$, as required.

We have built, from the mosaic $\mathbf{\Sigma} = (\rho, R, \kappa)$, the weft $\mathcal{E}xpl(\mathbf{\Sigma}) = \langle \kappa \circledR_\rho R\rangle$, which has the same set-valued realizations:

$$\mathrm{Real}_{\mathcal{A}mbi}(\mathcal{E}xpl(\mathbf{\Sigma}), \mathcal{S}et) \cong \mathrm{Real}_{\mathcal{A}mbi}(\mathbf{\Sigma}, \mathcal{S}et) \ .$$

5.10 Conclusion

This example, in contrast to the one in 2, is totally satisfactory: thanks to a mosaic and a ribbon product, it has been possible to meet our requirements.

Now, let us see what happens when dealing with a binary product constraint (in 5.11) and new deduction rules (in 5.12).

5.11 Product constraint

Let us replace the weft of ambigraphs $\mathbf{S}$ by $\mathbf{S}_+$, as in 2.11.

First we consider the constraint of product, and we say that the interpretation of the arrows π_1 and π_2 should satisfy $K(\mathtt{Ar}, ok)$, as well as each arrow in each ambigraph of the constraint. Of course, points should satisfy $K(\mathtt{Pt}, Err)$ and equations $K(\mathtt{Eq}, eq)$. Then it is easy to see that in each realization of $\mathbf{S}_+$ towards $\Theta \circ \kappa$, the interpretation of N^2 is $(\mathbb{N}^2)' = \mathbb{N}^2 \sqcup \{\varepsilon_{\mathbb{N}^2}\}$, as required.

$$\begin{tikzcd}
 & & & Y \arrow[r] \arrow[dlll] \arrow[drrr, "f_2'"] \arrow[dd] & Y' \arrow[dlll, "f_1'"'] \arrow[drrr] \arrow[dd, "f'"] & \{\varepsilon_Y\} \arrow[l] \arrow[dlll] \arrow[drrr] \arrow[dd] & & & \\
\mathbb{N} \arrow[r] & \mathbb{N}' & \{\varepsilon_{\mathbb{N}}\} \arrow[l] & & & & \mathbb{N} \arrow[r] & \mathbb{N}' & \{\varepsilon_{\mathbb{N}}\} \arrow[l] \\
 & & & \mathbb{N}^2 \arrow[r] \arrow[ulll] \arrow[urrr] & (\mathbb{N}^2)' \arrow[ulll] \arrow[urrr] & \{\varepsilon_{\mathbb{N}^2}\} \arrow[l] \arrow[ulll] \arrow[urrr] & & &
\end{tikzcd}$$

Then it follows from the equations that the interpretation of a is the addition $+$.

In this way, the mosaic $\mathbf{\Sigma}$ is extended into a mosaic $\mathbf{\Sigma}_+$ such that natural numbers with a predecessor operation correspond to:

1. a set-valued realization of $\mathbf{\Sigma}_+$,
2. or, equivalently, a set-valued realization of $\mathcal{E}xpl(\mathbf{\Sigma}_+)$.

This is wholly satisfactory from the point of view of realizations, however we will see that it does not behave very well from the point of view of programs (the notion of program will be studied in [guide3]).

Indeed, our choice for the properties of the interpretation of the product constraint has the following consequence. Given two terms $t_1 : S \to N$ and $t_2 : S \to N$ of $\mathbf{S}^+$ (for any point S of $\mathbf{S}^+$), the term:

$$t = fact(t_1, t_2) : S \to N^2$$

is defined *only* when t_1 and t_2 satisfy $K(\mathtt{Ar}, ok)$. However, usually the term t is meaningful as soon as t_1 and t_2 satisfy $K(\mathtt{Ar}, for)$.

Indeed, let $Y' = Y \sqcup \{\varepsilon_Y\}$ be a set with an error element, and let $f'_1 : Y' \to \mathbb{N}$ and $f'_2 : Y' \to \mathbb{N}$ be two maps which propagate the error (and which may create an error). We define the map $f' : Y' \to (\mathbb{N}^2)'$ by: $f'(\varepsilon_Y) = \varepsilon_{\mathbb{N}^2}$, and for all $y \in Y$:

- $f'(y) = \varepsilon_{\mathbb{N}^2}$ if $f'_1(y) = \varepsilon_{\mathbb{N}}$ or $f'_2(y) = \varepsilon_{\mathbb{N}}$,
- $f'(y) = (x_1, x_2)$ if $f'_1(y) = x_1 \in \mathbb{N}$ and $f'_2(y) = x_2 \in \mathbb{N}$.

Then, if $p'_1 : (\mathbb{N}^2)' \to \mathbb{N}'$ and $p'_2 : (\mathbb{N}^2)' \to \mathbb{N}'$ denote the two projections, the equalities $p'_1 \circ f' = f'_1$ and $p'_2 \circ f' = f'_2$ are partially satisfied on Y'. Precisely, if $def(f')$ denotes the set of elements y' of Y' such that $f'(y')$ is not an error, we see that the map f' is characterized by:

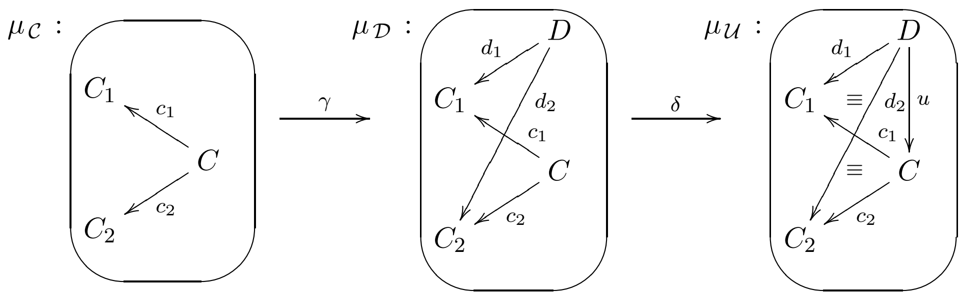
$$\begin{cases} def(f') = def(f'_1) \cap def(f'_2) \\ \text{and} \\ \forall y' \in def(f') \ : \ p'_1 \circ f'(y') = f'_1(y') \text{ and } p'_2 \circ f'(y') = f'_2(y') \ . \end{cases}$$

This means, if $f'_{l,i} = p'_i \circ f'$ and $f'_{r,i} = f'_i$ for $i = 1$ and 2, that:

$$\begin{cases} def(f'_{l,1}) \cap def(f'_{l,2}) = def(f'_{r,1}) \cap def(f'_{r,2}) \\ \text{and for all } y' \text{ in this set:} \\ f'_{l,1}(y') = f'_{r,1}(y') \text{ and } f'_{l,2}(y') = f'_{r,2}(y') \ . \end{cases}$$

We will see that it is possible to extend ρ in order to express this property of the product.

The potential of the constraint of product is:



and its body is defined by:

$$(C_1 \xleftarrow{c_1} C \xrightarrow{c_2} C_2) \quad \mapsto \quad (N \xleftarrow{\pi_1} N^2 \xrightarrow{\pi_2} N) \; .$$

The arrows c_1 and c_2 should satisfy $K(\mathtt{Ar}, ok)$ since the projections p'_1 and p'_2 are ok-maps. On the other hand, the arrows d_1 and d_2 should only satisfy $K(\mathtt{Ar}, for)$ since their interpretations f'_1 and f'_2 are any maps which propagate the error. Similarly, the arrow u should only satisfy $K(\mathtt{Ar}, for)$, since its interpretation f' has no other property, generally, than the propagation of the error. Now, we introduce a new comment for the equations $c_1 \circ u \equiv d_1$ and $c_2 \circ u \equiv d_2$ of $\mu_{\mathcal{U}}$.

First, both equations $c_1 \circ u \equiv d_1$ and $c_2 \circ u \equiv d_2$, which have the same domain D, are considered as a unique ingredient of $\mu_{\mathcal{U}}$. For this purpose, we add to **E** (this is a conservative extension):

- a composed arrow $\mathtt{domeq} = \mathtt{dom} \circ (\mathtt{p}_g \circ \mathtt{j}_{\mathtt{Eq}}) = \mathtt{dom} \circ (\mathtt{p}_d \circ \mathtt{j}_{\mathtt{Eq}}) : \mathtt{Eq} \to \mathtt{Pt}$ in order to get direct access to the common domain of both sides of an equation,
- a point PEq,
- two arrows $\mathtt{proj}_1 : \mathtt{PEq} \to \mathtt{Eq}$ and $\mathtt{proj}_2 : \mathtt{PEq} \to \mathtt{Eq}$,
- and the distinguished projective cone $\Gamma_{\mathtt{PEq}}$ which says that PEq should be interpreted as the set of pairs of equations with the same domain:

$$\Gamma_{\mathtt{PEq}} : \quad \begin{tikzcd} & \mathtt{PEq} \arrow[dl, "\mathtt{proj}_1"'] \arrow[dr, "\mathtt{proj}_2"] & \\ \mathtt{Eq} \arrow[dr, "\mathtt{domeq}"'] & & \mathtt{Eq} \arrow[dl, "\mathtt{domeq}"] \\ & \mathtt{Pt} & \end{tikzcd}$$

- in addition, in order to be able to speak about the two pairs of arrows with the same rank which are part of each pair of equations with the same domain, we also add to **E** the composed arrows:
  $\mathtt{pc}_1 = \mathtt{j}_{\mathtt{Eq}} \circ \mathtt{proj}_1 : \mathtt{PEq} \to \mathtt{RankP}$
  $\mathtt{pc}_2 = \mathtt{j}_{\mathtt{Eq}} \circ \mathtt{proj}_2 : \mathtt{PEq} \to \mathtt{RankP}$.

Then the pair $(c_1 \circ u \equiv d_1, c_2 \circ u \equiv d_2)$ belongs to the set $\mu_{\mathcal{U}}(\mathtt{PEq})$: it is a pair of equations of $\mu_{\mathcal{U}}$ with the same domain.

Now we introduce a new comment:

- $K(\mathtt{PEq}, ceq)$: consider a pair of equations with the same domain $(s_{l,1} \equiv s_{l,2}\, , \, s_{r,1} \equiv s_{r,2})$. Let $f'_{l,i}$ denote the interpretation of $s_{l,i}$ and $f'_{r,i}$ the interpretation of $s_{r,i}$ for $i = 1$ and 2. The interpretation of $(s_{l,1} \equiv s_{l,2}\, , \, s_{r,1} \equiv s_{r,2})$, satisfies the comment $K(\mathtt{PEq}, ceq)$ if:

$$\begin{cases} def(f'_{l,1}) \cap def(f'_{l,2}) = def(f'_{r,1}) \cap def(f'_{r,2}) \\ \text{and for all } y' \text{ in this set:} \\ f'_{l,1}(y') = f'_{r,1}(y') \text{ and } f'_{l,2}(y') = f'_{r,2}(y') \; . \end{cases}$$

Then the equations $s_{l,1} \equiv s_{l,2}$ and $s_{r,1} \equiv s_{r,2}$ generally do *not* satisfy the comment $K(\texttt{Eq}, eq)$.

Clearly, the property of the product can now be stated as follows: the interpretation of the pair of equations of $\mu_{\mathcal{U}}$ with the same domain ($c_1 \circ u \equiv d_1, c_2 \circ u \equiv d_2$) should satisfy the comment $K(\texttt{PEq}, ceq)$.

This is why we extend $\mathbf{F}$ (in a non-conservative way) and ρ, by adding:

- above $\texttt{PEq}$: a point $[\texttt{PEq}, ceq]$,
- above $\texttt{pc}_i$ (for $i = 1$ and 2): an arrow $[\texttt{pc}_i, ceq] : [\texttt{PEq}, ceq] \to [\texttt{RankP}, (for, for)]$.

Finally, in order to formalize the comment $K(\texttt{PEq}, ceq)$, we add to κ:

$\mathcal{A}$-Weft $\kappa([\texttt{PEq}, ceq])$:

lines:	$\kappa([\texttt{Pt}, Err])_0\,,\ \kappa([\texttt{Pt}, Err])_1\,,\ \kappa([\texttt{Pt}, Err])_2\,,$
point:	$\widehat{H}\,,$
arrows:	$k'_{g,1} : H'_0 \to H'_1\,,\ k^e_{g,1} : H^e_0 \to H^e_1\,,\ k'_{g,2} : H'_0 \to H'_2\,,\ k^e_{g,2} : H^e_0 \to H^e_2\,,$
	$k'_{d,1} : H'_0 \to H'_1\,,\ k^e_{d,1} : H^e_0 \to H^e_1\,,\ k'_{d,2} : H'_0 \to H'_2\,,\ k^e_{d,2} : H^e_0 \to H^e_2\,,$
	$\widehat{h}_{g,0} : \widehat{H} \to H'_0\,,\ \widehat{h}_{g,1} : \widehat{H} \to H'_1\,,\ \widehat{h}_{g,2} : \widehat{H} \to H'_2\,,$
	$\widehat{h}_{d,0} : \widehat{H} \to H'_0\,,\ \widehat{h}_{d,1} : \widehat{H} \to H'_1\,,\ \widehat{h}_{d,2} : \widehat{H} \to H'_2\,,$
equations:	$k'_{g,1} \circ h^e_0 \equiv h^e_1 \circ k^e_{g,1}\,,\ k'_{g,2} \circ h^e_0 \equiv h^e_2 \circ k^e_{g,2}\,,$
	$k'_{d,1} \circ h^e_0 \equiv h^e_1 \circ k^e_{d,1}\,,\ k'_{d,2} \circ h^e_0 \equiv h^e_2 \circ k^e_{d,2}\,,$
	$k'_{g,1} \circ h_0 \circ \widehat{h}_{g,0} \equiv h_1 \circ \widehat{h}_{g,1}\,,\ k'_{g,2} \circ h_0 \circ \widehat{h}_{g,0} \equiv h_2 \circ \widehat{h}_{g,2}\,,$
	$k'_{d,1} \circ h_0 \circ \widehat{h}_{d,0} \equiv h_1 \circ \widehat{h}_{d,1}\,,\ k'_{d,2} \circ h_0 \circ \widehat{h}_{d,0} \equiv h_2 \circ \widehat{h}_{d,2}\,,$
	$\widehat{h}_{g,0} \equiv \widehat{h}_{d,0}\,,\ \widehat{h}_{g,1} \equiv \widehat{h}_{d,1}\,,\ \widehat{h}_{g,2} \equiv \widehat{h}_{d,2}\,,$
c.p.d.:	$\widehat{\Gamma}_g\,,\ \widehat{\Gamma}_d\,.$

$$\kappa([\texttt{PEq}, ceq]) :\quad
\begin{tikzcd}
 & H_1 \arrow[r, "h_1"] & H'_1 = H_1 + H^e_1 & H^e_1 = \mathbb{1} \arrow[l, "h^e_1"'] \\
\widehat{H} \arrow[ur, bend left, "\widehat{h}_{g,1}"] \arrow[ur, bend right, "\widehat{h}_{d,1}"'] \arrow[ur, phantom, "\equiv"] \arrow[r, bend left, "\widehat{h}_{g,0}"] \arrow[r, bend right, "\widehat{h}_{d,0}"'] \arrow[r, phantom, "\equiv"] \arrow[dr, bend left, "\widehat{h}_{g,2}"] \arrow[dr, bend right, "\widehat{h}_{d,2}"'] \arrow[dr, phantom, "\equiv"] & H_0 \arrow[r, "h_0"] \arrow[ur, phantom, "\equiv\equiv"] \arrow[dr, phantom, "\equiv\equiv"] & H'_0 = H_0 + H^e_0 \arrow[u, bend left, "k'_{g,1}"] \arrow[u, bend right, "k'_{d,1}"'] \arrow[d, bend right, "k'_{g,2}"'] \arrow[d, bend left, "k'_{d,2}"] & H^e_0 = \mathbb{1} \arrow[l, "h^e_0"'] \arrow[u, bend left, "k^e_{g,1}"] \arrow[u, bend right, "k^e_{d,1}"'] \arrow[d, bend right, "k^e_{g,2}"'] \arrow[d, bend left, "k^e_{d,2}"] \arrow[ul, phantom, "\equiv\equiv"] \arrow[dl, phantom, "\equiv\equiv"] \\
 & H_2 \arrow[r, "h_2"] & H'_2 = H_2 + H^e_2 & H^e_2 = \mathbb{1} \arrow[l, "h^e_2"']
\end{tikzcd}$$

where the distinguished projective cones $\widehat{\Gamma}_g$ and $\widehat{\Gamma}_d$ are:

$$\widehat{\Gamma}_g :\quad
\begin{tikzcd}
 & H_1 \arrow[r, "h_1"] & H'_1 \\
\widehat{H} \arrow[ur, "\widehat{h}_{g,1}"] \arrow[r, "\widehat{h}_{g,0}"] \arrow[dr, "\widehat{h}_{g,2}"'] & H_0 \arrow[r, "h_0"] & H'_0 \arrow[u, "k'_{g,1}"'] \arrow[d, "k'_{g,2}"] \\
 & H_2 \arrow[r, "h_2"] & H'_2
\end{tikzcd}
\qquad
\widehat{\Gamma}_d :\quad
\begin{tikzcd}
 & H_1 \arrow[r, "h_1"] & H'_1 \\
\widehat{H} \arrow[ur, "\widehat{h}_{d,1}"] \arrow[r, "\widehat{h}_{d,0}"] \arrow[dr, "\widehat{h}_{d,2}"'] & H_0 \arrow[r, "h_0"] & H'_0 \arrow[u, "k'_{d,1}"'] \arrow[d, "k'_{d,2}"] \\
 & H_2 \arrow[r, "h_2"] & H'_2
\end{tikzcd}$$

and for $i = 1$ and 2:

Homomorphism $\kappa([\mathtt{pc}_i, \mathit{ceq}]) : \kappa([\mathtt{RankP}, (\mathit{for}, \mathit{for})]) \to \kappa([\mathtt{PEq}, \mathit{ceq}])$:
lines: $\kappa([\mathtt{Pt}, \mathit{Err}])_1 \mapsto \kappa([\mathtt{Pt}, \mathit{Err}])_0\,,\ \kappa([\mathtt{Pt}, \mathit{Err}])_2 \mapsto \kappa([\mathtt{Pt}, \mathit{Err}])_i\,,$
arrows: $k'_g \mapsto k'_{g,i}\,,\ k^e_g \mapsto k^e_{g,i}\,,\ k'_d \mapsto k'_{d,i}\,,\ k^e_d \mapsto k^e_{d,i}\ .$

Now, let:

$$\nu_{\mathcal{U}}([\mathtt{PEq}, \mathit{ceq}]) = \mu_{\mathcal{U}}(\mathtt{PEq}) = \{(c_1 \circ u \equiv d_1, c_2 \circ u \equiv d_2)\}\ .$$

In this way, $\mathit{fact}(t_1, t_2)$ is defined as required: given two terms $t_1 : T \to N$ and $t_2 : T \to N$ of $\mathbf{S}_+$, the term $t = \mathit{fact}(t_1, t_2) : T \to N^2$ is defined once t_1 and t_2 satisfy $K(\mathtt{Ar}, \mathit{for})$. However, of course, if f'_1, f'_2 and f' denote the interpretations of t_1, t_2 and t, generally $p'_1 \circ f'$ is *different* from f'_1, and $p'_2 \circ f'$ is *different* from f'_2. We only have, as required, the *partial* equalities:

$$p'_1 \circ f'(y') = f'_1(y') \text{ and } p'_2 \circ f'(y') = f'_2(y') \text{ for all } y' \in \mathit{def}(f') = \mathit{def}(f'_1) \cap \mathit{def}(f'_2)\ .$$

5.12 Other deduction rules

Let us come back to the ribbon product $\langle \kappa \circledR_\rho R \rangle$ as in 5.8.

From the information we have about the interpretations of p and s, we may deduce that the interpretation of $p \circ s$ propagates the error, nothing more. On the other hand, the interpretation of id_N is an ok-map. Since the equation $p \circ s \equiv id_N$ is part of $\mathbf{S}$, clearly the interpretation of $p \circ s$ is also an ok-map. We will see how it is possible to extend $\mathbf{F}$ and ρ in order to express the deduction rule:

- *if the interpretation of the right-hand side of an equation is an ok-map, then the interpretation of its left-hand side is also an ok-map.*

This will give an example of the way a distinguished projective cone in $\mathbf{F}$ can express a conjunction of properties.

The hypothesis:

- *the interpretation of the right-hand side of an equation is an ok-map*

is a new property, which corresponds to a new point of $\mathbf{F}$. A priori, if the interpretation of an equation satisfies $K(\mathtt{Eq}, \mathit{eq})$, then the interpretation of its right-hand side satisfies $K(\mathtt{Ar}, \mathit{for})$, which is weaker than $K(\mathtt{Ar}, \mathit{ok})$. So, we have to express the conjunction of the properties $K(\mathtt{Eq}, \mathit{eq})$ (for an equation) and $K(\mathtt{Ar}, \mathit{ok})$ (for its right-hand side), in a way compatible with $K(\mathtt{Ar}, \mathit{for})$ (for its right-hand side).

First of all, we add to $\mathbf{E}$ (this is a conservative extension):

- the identity arrow $id_{\mathtt{Eq}}$,
- the composed arrows (in order to access directly to both members of a equation):
  $\mathtt{q}_l = \mathtt{p}_l \circ \mathtt{j}_{\mathtt{Eq}} : \mathtt{Eq} \to \mathtt{Ar}$
  $\mathtt{q}_r = \mathtt{p}_r \circ \mathtt{j}_{\mathtt{Eq}} : \mathtt{Eq} \to \mathtt{Ar}$
- and the distinguished projective cone:

$$\begin{tikzcd}
 & \mathtt{Eq} \arrow[dl, "id_{\mathtt{Eq}}"'] \arrow[dr, "\mathtt{q}_d"] & \\
\mathtt{Eq} \arrow[dr, "\mathtt{q}_d"'] & & \mathtt{Ar} \arrow[dl, "id_{\mathtt{Ar}}"] \\
 & \mathtt{Ar} &
\end{tikzcd}$$

Then, in order to express the hypothesis, we add to **F**:

- above `Eq`: a point [`Eq`, eq_d].
- above $id_{\texttt{Eq}}$: an arrow $[id_{\texttt{Eq}}, eq_d \Rightarrow eq] : [\texttt{Eq}, eq_d] \to [\texttt{Eq}, eq]$.
- above $\texttt{q}_d$: an arrow $[\texttt{q}_d, eq_d \Rightarrow ok] : [\texttt{Eq}, eq_d] \to [\texttt{Ar}, ok]$.
- above the distinguished projective cone just added to **E**: the distinguished projective cone:

$$\begin{tikzcd}
 & {[\texttt{Eq}, eq_d]} \arrow[dl, "{[id_{\texttt{Eq}}, eq_d \Rightarrow eq]}"'] \arrow[dr, "{[\texttt{q}_d, eq_d \Rightarrow ok]}"] & \\
{[\texttt{Eq}, eq]} \arrow[dr, "{[\texttt{q}_d, eq]}"'] & & {[\texttt{Ar}, ok]} \arrow[dl, "{[id_{\texttt{Ar}}, ok \Rightarrow for]}"] \\
 & {[\texttt{Ar}, for]} &
\end{tikzcd}$$

Finally, in order to express the deduction rule, we add to **F**:

- above $\texttt{q}_g$: an arrow $[\texttt{q}_g, eq_d \Rightarrow ok] : [\texttt{Eq}, eq_d] \to [\texttt{Ar}, ok]$.

Now let us add to the counter-model κ:

$\mathcal{A}$-Weft $\kappa([\texttt{Eq}, eq_d])$:
extends: $\kappa([\texttt{Eq}, eq])$,
arrow: $k_d : H_1 \to H_2$,
equation: $k'_d \circ h_1 \equiv h_2 \circ k_d$.

$$\kappa([\texttt{Eq}, eq_d]) : \quad \begin{tikzcd}
H_1 \arrow[r, "h_1"] \arrow[d, "k_d"] \arrow[dr, phantom, "\equiv"] & H'_1 = H_1 + H^e_1 \arrow[d, shift right, "k'_g"'] \arrow[d, shift left, "k'_d"] \arrow[d, phantom, "\equiv"] \arrow[dr, phantom, "\equiv\equiv"] & H^e_1 = \mathbb{1} \arrow[l, "h^e_1"'] \arrow[d, shift right, "k^e_g"'] \arrow[d, shift left, "k^e_d"] \arrow[d, phantom, "\equiv"] \\
H_2 \arrow[r, "h_2"] & H'_2 = H_2 + H^e_2 & H^e_2 = \mathbb{1} \arrow[l, "h^e_2"']
\end{tikzcd}$$

Homomorphism $\kappa([id_{\texttt{Eq}}, eq_d \Rightarrow eq]) : \kappa([\texttt{Eq}, eq]) \to \kappa([\texttt{Eq}, eq_d])$:
it is the extension.
Homomorphism $\kappa([\texttt{q}_d, eq_d \Rightarrow ok]) : \kappa([\texttt{Ar}, ok]) \to \kappa([\texttt{Eq}, eq_d])$:
lines: $\kappa([\texttt{Pt}, Err])_1 \mapsto \kappa([\texttt{Pt}, Err])_1$, $\kappa([\texttt{Pt}, Err])_2 \mapsto \kappa([\texttt{Pt}, Err])_2$,
arrows: $k \mapsto k_d$, $k' \mapsto k'_d$, $k^e \mapsto k^e_d$.
Homomorphism $\kappa([\texttt{q}_g, eq_d \Rightarrow ok]) : \kappa([\texttt{Ar}, ok]) \to \kappa([\texttt{Eq}, eq_d])$:
lines: $\kappa([\texttt{Pt}, Err])_1 \mapsto \kappa([\texttt{Pt}, Err])_1$, $\kappa([\texttt{Pt}, Err])_2 \mapsto \kappa([\texttt{Pt}, Err])_2$,
arrows: $k \mapsto k_d$, $k' \mapsto k'_g$, $k^e \mapsto k^e_g$.

Actually, for $\kappa([\texttt{q}_g, eq_d \Rightarrow ok])$ to be really a homomorphism, we slightly extend $\kappa([\texttt{Eq}, eq_d])$. More precisely, we add the equation $k'_g \circ h_1 \equiv h_2 \circ k_d$. This extension is conservative, because this equation can be deduced from $k'_d \circ h_1 \equiv h_2 \circ k_d$ and $k'_g \equiv k'_d$.

In this way, if we know that $(p \circ s \equiv id_N)$ satisfies $K(\texttt{Eq}, eq)$ and that id_N satisfies $K(\texttt{Ar}, ok)$, we may deduce that:

$$p \circ s \text{ satisfies } K(\texttt{Ar}, ok)\ .$$

6 Conclusion

In this paper, using the basic notions introduced in [guide1], we have defined mosaics and the ribbon product, and studied one of their applications.

Whereas simple specifications have been defined in [guide1, section 2.5] as *wefts*, more complicated specifications are defined in 4.3 as *mosaics*. Their interest lies in their ability to deal with the implicit features of computer languages.

The ribbon product is a new constructor for specifications. The fundamental result regarding it states that each mosaic can be made explicit thanks to a ribbon product. This result justifies the use of mosaics, and allows us to *avoid* making them explicit.

In this paper, we focus on the realizations of a mosaic. From this point of view, the apparent weft of a mosaic is rather uninteresting: its realizations are irrelevant, and the apparent weft must be corrected, thanks to a stratification, before yielding something meaningful. On the contrary, in [guide3] we will focus on the major role of the apparent weft in the design of programs.

Thanks to the ribbon product, we have been able to prove that the so-called explicit and implicit points of view about specifications in computer science, far from being opposed, are strongly related. In [guide3], this will enable us to build a common framework for dealing with functional and imperative programs, which preserves the specificities of both programming modes.

References

[ref] Christian Lair and Dominique Duval. Sketches and specifications: reference manual. *Rapport de recherche du LACO*, 2000. http://www.unilim.fr/laco/rapports.

[ref1] First part: Compositive graphs.

[ref2] Second part: Projective sketches.

[ref3] Third part: Models.

[guide] Dominique Duval and Christian Lair. Sketches and specifications: User's guide. *Rapport de recherche du LACO*, 2000. http://www.unilim.fr/laco/rapports.
Also: D. Duval. *idem*. Submitted for publication.

[guide1] First part: Wefts for explicit specification.

[guide2] Second part: Mosaics for implicit specification. This paper.

[guide3] Third part: Functional and imperative programs. Work in progress.

[state] Dominique Duval and Christian Lair. Mosaics for specifications with implicit state. *Rapport de recherche du LACO*, 2000. http://www.unilim.fr/laco/rapports.
Also: D. Duval. *idem*. Submitted for publication.

[Astesiano *et al.* 99] E. Astesiano, H.-J. Kreowski, and B. Krieg-Brckner. *Algebraic Foundations of Systems Specification*. Springer, 1999.

[Astesiano & Zucca 95] Egidio Astesiano and Elena Zucca. D-oids: a model for dynamic datatypes. *Mathematical Structures in Computer Science*, 5, pp. 257–282, 1995.

[Dauchy & Gaudel 94] Pierre Dauchy and Marie-Claude Gaudel. Algebraic specifications with implicit state. Rapport de Recherche 887, Université de Paris-Sud, Laboratoire de Recherche en Informatique, 1994.

[Ehresmann 66] Charles Ehresmann. Introduction to the theory of structured categories. Technical Report 10, University of Kansas at Lawrence, 1966.

[Ehresmann 68] Charles Ehresmann. Esquisses et types de structures algébriques. *Bulletin de l'Institut Polytechnique, Iaşi*, 14, pp. 1–32, 1968.

[Ehrig & Orejas 94] Hartmut Ehrig and Fernando Orejas. Dynamic abstract data types: An informal proposal. *Bulletin of the European Association for Theoretical Computer Science*, 53, pp. 162–169, 1994.

[Ehrig & Orejas 98] Hartmut Ehrig and Fernando Orejas. Integration paradigm for data types and process specification techniques. *Bulletin of the European Association for Theoretical Computer Science*, 65, pp. 90–97, 1998.

[Filliatre 99] Jean-Christophe Filliâtre. Proof of imperative programs in type theory. In *Proceedings of the TYPES'98 workshop*, Lecture Notes in Computer Science 1657. Springer-Verlag, 1998.

[Freyd 73] Peter J. Freyd. Properties invariant within equivalence types of categories. In *Eilenberg Festschrift*. Academic Press, 1973.

[Freyd & Scedrov 90] Peter J. Freyd and A. Scedrov. *Categories, Allegories.* Mathematics Library 39. North-Holland, 1990.

[Gaudel *et al.* 99] Marie-Claude Gaudel, Carole Khoury, and Alexandre Zamulin. Dynamic systems with implicit state. In *ETAPS'99*, Lecture Notes in Computer Science 1577, pp. 114–128. Springer-Verlag, 1999.

[Goguen 78] Joseph A. Goguen. Abstract errors for abstract data types. In E.J. Neuhold, editor, *Formal Description of Programming Concepts*, pp. 491–525. North-Holland, 1978.

[Goguen *et al.* 78] Joseph A. Goguen, J.W. Thatcher, and Eric G. Wagner. An initial algebra approach to the specification, correctness, and implementation of abstract data types. In R.T. Yeh, editor, *Current Trends in Programming Methodology, Volume IV: Data Structuring*, chapter 5, pp. 80–149. Prentice-Hall, 1978.

[Gurevich 91] Yuri Gurevich. Evolving algebras: A tutorial introduction. *Bulletin of the European Association for Theoretical Computer Science*, 43, pp. 264–284, 1991.

[Gurevich 99] Yuri Gurevich. The sequential ASM thesis. *Bulletin of the European Association for Theoretical Computer Science*, 67, pp. 93–124, 1999.

[Hopcroft & Ullman 79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to automata theory, languages and computation.* Addison-Wesley, 1979.

[Lafont 88] Yves Lafont. The linear abstract machine. *Theoretical Computer Science*, 59, pp. 157–180, 1988.

[Lair 87] Christian Lair. Trames et sémantiques catégoriques des systèmes de trames. *Diagrammes*, 18, pp. CL1–CL47, 1987.

[Lair 93] Christian Lair. Eléments de théorie des patchworks. *Diagrammes*, 29, pp. CL1–CL29, 1993.

[Lellahi & Zamulin 99] Kazem Lellahi and Alexandre Zamulin. Dynamic system based on update sets. Rapport de Recherche 99-03, Université Paris-Nord, Institut Galilée, LIPN, 1999.

[Moggi 91] Eugenio Moggi. Notions of computation and monads. *Information and computation*, 93, pp. 55–92, 1991.

[Wadler 85] Philip Wadler. Monads for functional programming. In J. Jeuring and E. Meijer, editors, *Advanced Functional Programming, Proceedings of the Båstad Spring School 1985*, Lecture Notes in Computer Science 925. Springer-Verlag, 1985.

[Wadler 90] Philip Wadler. Linear types can change the world! In M. Broy and C. B. Jones, editors, *Programming Concepts and Methods.* North Holland, 1990.

[Wirsing 90] Martin Wirsing. Algebraic specifications. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, chapter 13, pp. 675–788. Elsevier, 1990.

Contents

Index